



UNIVERSITÀ DI PISA

DIPARTIMENTO DI INGEGNERIA DELL' INFORMAZIONE

Tesi di Laurea Magistrale in

INGEGNERIA DELLE TELECOMUNICAZIONI

**SVILUPPO SOFTWARE/HARDWARE DI UN MODEM ACUSTICO
SUBACQUEO A SPETTRO ESPANSO**

Candidato:

Claudio Tascone

Relatori:

Prof. Ruggero Reggiannini

Ing. Marco Moretti

Ing. Marco Michelini

Introduzione	5
Capitolo1: Strumentazione e Ambiente di sviluppo	6
1.1 Introduzione	6
1.2 Strumentazione	6
1.2.1 Chassis NI PXIe-1085 3U	8
1.2.2 Controller NI PXIe-8135	10
1.2.3 Scheda di acquisizione dati Serie X, NI PXIe-6361	11
1.3 Linguaggio di Programmazione	12
Capitolo2: Driver NI-DAQmx	16
2.1 Introduzione	16
2.2 NI-DAQmx	16
2.3 Canali fisici, Canali virtuali e Tasks	18
2.3.1 Canale fisico	18
2.3.2 Canale virtuale	18
2.3.3 Tasks	19
2.4 Funzioni DAQmx	19
2.4.1 NI-DAQmx Flatten Channel String	20
2.4.2 NI-DAQmx Create Virtual Channel	20
2.4.3 NI-DAQmx Timing	21
2.4.4 NI-DAQmx Write	22
2.4.5 NI-DAQmx Read	23
2.4.6 NI-DAQmx Start Task	23
2.4.7 NI-DAQmx Clear Task	23
2.4.8 NI-DAQmx Properties	24
Capitolo3: Sistema di trasmissione	25
3.1 Introduzione	25
3.2 Sistema di comunicazione digitale generico	26
3.3 Modem acustico	29
3.3.1 Codifica a protezione d'errore	30
3.3.2 Mappatura con codifica differenziale	30
3.3.3 Sagomatura in banda base	31
3.3.4 Espansione spettrale	32
3.3.5 Despredding	34
3.3.6 Stima del ritardo	37
3.3.6.1 Stima del raggio a maggiore potenza	38
3.3.6.2 Stima di più raggi	40
3.3.6.3 Trova massimi	41
3.3.6.4 Somma valori di correlazione	46
3.3.7 Stima di canale	48
3.3.8 MRC	49

Capitolo4: Analisi sulle prestazioni del modem	52
4.1 Introduzione	52
4.2 Analisi sull'energia e sul rumore	52
4.2.1 Energia	52
4.2.2 Rumore	54
4.2.3 Calcolo della BER al variare di E_b/N_0 dB	55
4.3 Impostazioni di progetto	55
4.4 Simulazioni software	56
4.4.1 Primo scenario	57
4.4.2 Secondo scenario	61
4.4.2.1 Prestazioni al variare della lunghezza del preambolo	64
4.4.3 Terzo scenario: condizione su Tsw non rispettata	66
4.4.3.1 Prima simulazione	67
4.4.3.2 Seconda simulazione	69
4.4.4 Quarto scenario	70
 Capitolo5: Prototipo trasmettitore modem acustico	 74
5.1 Introduzione	74
5.2 Codice per la gestione della scheda	74
5.3 Hardware utilizzato nel progetto	75
5.3.1 Misure dello spettro	78
 Sitografia	 82

GLOSSARIO

- LV = LabView
- PXI = PCI eXtensions for Instrumentation
- NI = National Instruments
- DAQ = Data acquisition
- VI = Virtual instrument
- MAX = Measurement & Automation Explorer
- AI = Analog Input
- AO = Analog Output
- MRC = Maximal Ratio Combining
- S2C = Modulazione Sweep Spread Carrier
- BER = bit error rate

INTRODUZIONE

La crescente esigenza di scambiare informazioni, obbliga a trovare evoluzioni tecnologiche in ambito informatico e telecomunicazionistico. Questo ha portato a una digitalizzazione dei segnali da trasmettere in tutti gli ambiti della comunicazione: telefonia, radio, internet, TV, etc., che rispetto ai segnali analogici hanno notevoli vantaggi, soprattutto in termini di banda occupata e qualità del segnale. Per questo motivo, con l'utilizzo di un'adeguata strumentazione (chassis + controller + scheda di acquisizione) e di uno specifico linguaggio di programmazione (LabVIEW), si è voluto sviluppare un sistema di comunicazione digitale *software/hardware*.

Nella seguente tesi, dapprima si fa una panoramica di un sistema di comunicazione digitale classico: sistema lineare con utilizzo di modulazioni M-QAM e M-PSK; in seguito viene descritto in maniera dettagliata il funzionamento di un MODEM ACUSTICO SUBACQUEO.

Nel primo capitolo è data una descrizione della strumentazione e del linguaggio di programmazione utilizzati; nel secondo, sono descritte le funzioni utilizzate per gestire la scheda di acquisizione dati; nel terzo, viene data una descrizione teorica di un sistema di trasmissione digitale generico e quella del modem acustico, soffermandosi soprattutto su quest'ultimo; nel quarto, si analizzano le prestazioni del modem acustico in termini di bit error rate (BER); nel quinto, e ultimo capitolo, si dà una breve descrizione su come mettere assieme le funzioni DAQmx per gestire la scheda di acquisizione ed inoltre vengono mostrati sull'analizzatore di spettro i grafici relativi al segnale inviato dalla scheda.

CAPITOLO 1: STRUMENTAZIONE E AMBIENTE DI SVILUPPO

1.1 INTRODUZIONE

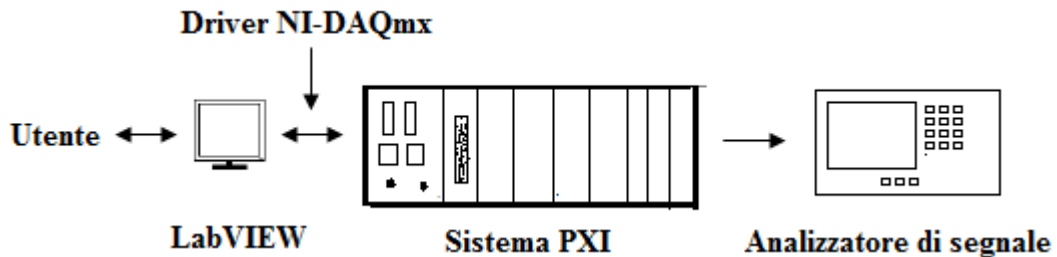


Fig. 1.1: schema logico sistema di misura

La strumentazione e l'ambiente di sviluppo sono i due elementi principali per la progettazione di sistemi di misura che permettono la trasmissione e l'acquisizione di dati e la relativa elaborazione del segnale. In particolare, come rappresentato in figura 1.1, il sistema in esame è realizzato con *LabVIEW* (LV) e un *sistema PXI* in cui la loro interazione è possibile grazie al *driver DAQmx*. Questa loro combinazione ha permesso di realizzare un sistema che acquisisce segnali reali con relativa elaborazione e genera segnali reali con la possibilità di mostrarli su di un oscilloscopio o di ricavarne lo spettro tramite un analizzatore di spettro. Per questo, di seguito sarà descritta l'apparecchiatura e il linguaggio di programmazione utilizzati nella progettazione del sistema di comunicazione hardware digitale.

1.2 STRUMENTAZIONE

La strumentazione utilizzata si basa sul sistema **PXI** (PCI eXtensions for Instrumentation) che è una piattaforma basata su PC in grado di offrire una soluzione di distribuzione a elevate prestazioni e costi ridotti per i sistemi di misura e automazione. PXI combina le funzioni di bus elettrico PCI con il pacchetto modulare e robusto Eurocard di CompactPCI e aggiunge bus di sincronizzazione specializzati e funzioni software. In seguito si è introdotto PXI Express (si è integrato il bus PCI Express) e questo ha portato a maggiori velocità del bus. I sistemi PXI sono costituiti da tre elementi fondamentali: Chassis PXI, Controller di sistema e Moduli di periferica.

- **Chassis PXI:** offre un alloggiamento robusto e modulare per il sistema. Lo chassis PXI fornisce i bus di comunicazione PCI e PCI Express per il controller e i moduli. Il bus PCI utilizza una topologia di bus condivisa il cui picco di bit-rate teorica di 132 MByte/s è diviso tra più dispositivi in modo che i diversi dispositivi possano comunicare sul bus. PCI Express ha una topologia di bus point-to-point che sostituisce il bus condiviso con uno switch condiviso. Questo fornisce a ogni dispositivo l'accesso diretto al bus e di conseguenza anche alle proprie pipeline di dati chiamate **lane**. È possibile raggruppare questi percorsi per aumentare il bit-rate per ottenere trasferimento dati fino a 4 GB/s. Inoltre sono presenti i bus di Triggering e Temporizzazione.

Le dimensioni dello chassis sono 3U e 6U (U = rack Unit, è un'unità di misura utilizzata per indicare l'altezza dei componenti installati in un rack, che è un sistema standard d'installazione fisica di componenti hardware a scaffale). La strumentazione modulare PXI aggiunge un clock di riferimento di sistema dedicato a 10 MHz, PXI Trigger bus, star Trigger bus e un bus locale slot-to-slot per temporizzazione avanzata e sincronizzazione.

- **Controller PXI**

Tutti gli chassis PXI includono uno slot per il controller di sistema nel primo alloggiamento sulla parte sinistra (slot 1). Vi sono vari tipi di controller PXI, come: controller embedded a elevate prestazioni con sistemi operativi Microsoft Windows oppure real-time (NI LabVIEW Real-Time), e controller remoti da computer desktop, workstation, server o laptop. Le due opzioni di controller sono controllo PXI da laptop e controllo PXI da PC. In particolare i controller Embedded eliminano la necessità di un PC esterno, perché forniscono un sistema completo all'interno dello chassis PXI. Questi sono dispositivi standard dotati di CPU integrata, disco fisso, RAM, scheda Ethernet, porte USB etc. Invece, con il controllo di PXI da laptop è possibile controllare i sistemi

PXI direttamente da computer mediante una connessione completamente trasparente al software. All'avvio il computer laptop riconosce tutte le periferiche del sistema PXI come schede PCI, in questo modo è possibile controllare questi dispositivi tramite un portatile.

- **Moduli di periferica**

Sono delle schede che vengono inserite negli slot e sono gestite dal controller. Vi sono disponibili moltissimi moduli per una vasta gamma di applicazioni: elaborazione del segnale, RF, temporizzazione e sincronizzazione etc.

Di seguito vengono descritti i componenti utilizzati per lo sviluppo del sistema di misura.

1.2.1 Chassis NI PXIe-1085 3U:

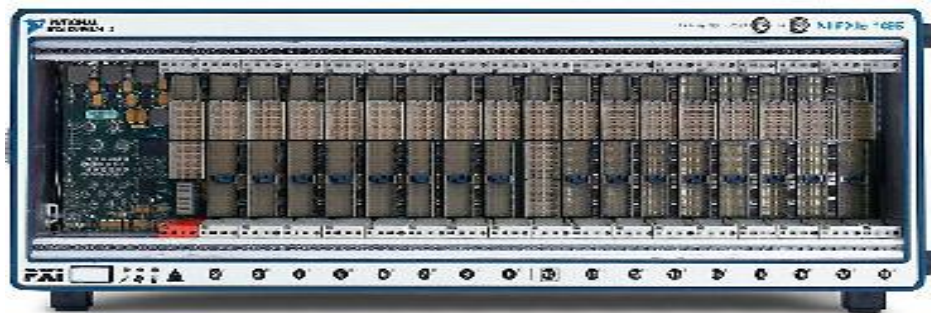


Fig. 1.2: Cestello PXIe-1085 3U

Le caratteristiche dello *Chassis NI PXIe-1085 3U* sono:

- 16 slot ibridi, 1 slot di temporizzazione di sistema PXI Express
- bit-rate dedicata di 4 GB/s per slot, 12 GB/s di Bit-Rate totale di sistema;
- Compatibilità con PXI, PXI Express, CompactPCI e moduli CompactPCI Express
- 925 W di potenza totale, 0-55° C senza perdita di prestazioni
- Porte Ethernet per monitorare lo stato dei componenti dello chassis
- Alimentazione rimovibile

PXIe-1085 ha 18 slot PCI Express Gen 2 ed integra un backplane interamente ibrido ad elevato bit-rate per lo sviluppo di una vasta gamma di applicazioni di test e misura. Il tipo di connettore ibrido in ogni slot periferico permette di ottenere la migliore flessibilità per il posizionamento dei moduli di strumentazione. PXIe-1085 include anche tutte le funzioni e specifiche PXI, incluso supporto per i moduli PXI e PXI Express con un clock di riferimento da 10 MHz, PXI trigger bus, PXI star trigger per moduli PXI e clock di riferimento a 100 MHz, SYNC100, e PXI star trigger differenziale per moduli PXI Express. NI PXIe-1085 integra la tecnologia PCI Express Generation 2, ed è in grado di fornire una Bit-Rate da 4GB/s per slot e una Bit-Rate totale di sistema di 12 GB/s tramite 2 collegamenti PCIe x8 Gen su ogni slot.

Lo chassis fornisce una potenza totale di 925W su tutto l'intervallo di temperatura da 0-55 °C senza perdita di prestazioni. Grazie a questo tipo di caratteristiche, il dispositivo è in grado di fornire una potenza di 38.25W per slot, in uno chassis completamente caricato. Inoltre include ventole inseribili nella parte posteriore dello chassis. L'alimentazione può essere rimossa singolarmente. Di seguito è riportata una tabella riguardante le caratteristiche degli slot.

SLOT	PXI Express System	HYBRID (PXI)
Bus di Segnalazione	PCI Express Gen 2 (1x16, 1x8)	PCI Express Gen 2 (x8) PCI (32/33)
Bit-Rate (teorica)	12 GB/s dedicati PXI Express. 132 MB/s condivisa per PXI	4 GB/s dedicato (PXI Exp). 132 MB/s condivisa per PXI
Numero di Slot	1	16

Tab 1.1: caratteristiche slot

Per i moduli PXI, il cestello è completamente conforme con gli standard PXI di temporizzazione e sincronizzazione. Lo chassis include un clock di riferimento a 10 MHz con un accuratezza di ± 25 parti per milione (ppm), un jitter inferiore a 5 ps e uno slot-to-slot skew massimo di 1 ns. Per i moduli PXI Exp, in aggiunta alle caratteristiche di temporizzazione e

sincronizzazione, la NI PXIe-1085 fornisce un clock di riferimento di 100 MHz con una precisione di ± 25 ppm, meno di 3 ps jitter e un massimo slot-to-slot skew di 100 ps.

1.2.2 Controller NI PXIe-8135



Fig. 1.3: Controller NI PXIe-8135

Le caratteristiche del *Controller NI PXIe-8135* sono:

- Processore 2.3 GHz quad-core Intel Core i7-3610QE (fino a 3.3 GHz in modalità Turbo Boost, single-core)
- Controller embedded PXI Express a elevata Bit-Rate con sistema fino a 8 GB/s e 2 GB/s di Bit-Rate per slot
- 4 GB (1 x 4 GB DIMM) dual-channel 1600 MHz DDR3 RAM standard, fino a 16 GB
- 2 SuperSpeed USB, 4 Hi-Speed USB, 2 schede Gigabit Ethernet, GPIB, seriale, e altre
- Diagnostica del disco rigido e della memoria In-ROM
- Sistema operativo Windows e driver già installati, ripristino basato su disco fisso
- Disco rigido 250 GB SATA 7200 rpm

NI PXIe-8135 è un controller embedded a prestazioni avanzate basato su processore quad-core Intel Core i7-3610QE da utilizzare nei sistemi PXI. Per aumentare il numero di thread che è possibile elaborare, si utilizza la tecnologia '**Intel Hyper-Threading**' in cui ogni core è visto come due core virtuali, per un totale di otto core virtuali. Questi otto core sono in grado di eseguire otto compiti di elaborazione, che è vantaggioso negli ambienti multitasking come

Windows 7. In ambienti di sviluppo multithreading, come LV, è possibile sfruttare al massimo i core di elaborazione separando i compiti in thread indipendenti. NI-8135 dispone di una base di 2.3 GHz di frequenza di clock e, con la tecnologia '**Intel Turbo Boost**', la frequenza aumenta automaticamente in base al tipo di applicazione. Ad esempio, durante l'esecuzione di applicazioni che generano un solo thread, la CPU pone i tre core inutilizzati in uno stato d'inattività e aumenta la frequenza di clock del nucleo attivo da 2.3 GHz a 3.3 GHz. L'utilizzo del processore i7-3610QE combinato al PCI Express, ha portato ad una Bit-Rate di 2 GB/s per slot ed una Bit-Rate complessiva di sistema fino a 8 GB/s. Molte delle porte di I/O sono integrate sul controller, quindi tutti gli slot dello chassis rimangono disponibili per i moduli di misura e controllo.

1.2.3 Scheda di acquisizione dati Serie X, NI PXIe-6361



Fig. 1.4: scheda di acquisizione PXIe-6361

Le caratteristiche della *Scheda NI PXIe-6361* sono:

- 16 ingressi analogici, 1 canale 2 MS/s, multicanale 1MS/s, risoluzione a 16 bit, ± 10 V
- Due uscite analogiche, 1 canale 2.86 MS/s, 2 canali 2 MS/s risoluzione a 16-bit, ± 10 V
- 24 linee I/O digitali (8 con temporizzazione hardware fino a 10 MHz)
- Quattro counter/timers a 32-bit per PWM (pulse width modulation), encoder, frequenza, conteggio eventi e altro
- Triggering analogico e digitale e temporizzazione avanzata con la tecnologia NI-STC3

- Supporto per Windows 7/Vista/XP/2000

I dispositivi DAQ multifunzione NI della Serie X permettono di ottenere maggiori prestazioni con un bus PCI Express ad alta velocità, tecnologia di temporizzazione e sincronizzazione NI-STC3 e driver ottimizzato per il multicore. La tecnologia NI-STC3 è un ASIC progettato per funzioni avanzate di temporizzazione, triggering e sincronizzazione. Questa tecnologia comprende:

- Quattro counter/timers
- Una base dei tempi a 100 MHz per un triggering (risposta) più rapida
- Time engine indipendenti per l'analogico e il digitale

In contrasto con PCI-to-PCI Express bridge chip, che limita la Bit-Rate del dispositivo a quella del bus PCI, i dispositivi PCI Express and PXI Express X Series usano un'interfaccia nativa x1 che fornisce fino a 250 MB/s in ogni direzione. Questa scheda è compatibile con il driver NI-DAQmx 9.0 o versioni successive. La NI PXIe-6361 include circuiti analogici, digitali e contatori per i tipi più comuni di misurazioni statistiche e forme d'onda. Con LV si possono facilmente acquisire i dati e visualizzarli su una varietà di grafici e display. Per l'acquisizione/trasmissione dei dati si utilizza un connettore VHDCI a 68 pin. Per tutte le altre informazioni visualizzare la documentazione fornita da National Instruments.

1.3 LINGUAGGIO DI PROGRAMMAZIONE

L'ambiente di sviluppo utilizzato è **LabVIEW** che è un linguaggio di programmazione **grafico** il cui codice è di tipo **iconografico**, anziché testuale: utilizza icone, terminali e collegamenti. Il codice è eseguito in maniera parallela (ad esempio, si possono avere due while loop ciascuno dei quali viene eseguito continuamente, senza mai terminare. In tal caso, il LV eseguirà, contemporaneamente, sia le istruzioni di uno che dell'altro). Inoltre offre integrazione hardware

e compatibilità ad ampio raggio con una vasta gamma di strumenti. Il punto di partenza per la realizzazione di un programma è l'apertura di una **VI** (Virtual Instrument) che è formata da due pannelli: Pannello di Controllo e Diagramma a Blocchi.

Il pannello di controllo (visibile allo sviluppatore e all'utente finale), di colore grigio (di default), costituisce un'interfaccia grafica creata dal programmatore con il quale l'utente finale può interagire per impostare i parametri di progetto e per visualizzare i risultati ottenuti. **Il diagramma a blocchi** (visibile solo allo sviluppatore), di colore bianco (di default), contiene delle funzioni base con le quali è possibile realizzare funzioni più complesse che infine si potranno assemblare per la realizzazione di un'applicazione finale. In entrambe le finestre è presente la **Palette dei Comandi**. Nel caso del pannello di controllo è chiamata **Palette dei Controlli** e contiene tutti i controlli (oggetto con cui l'utente interagisce per immettere valori) e gli indicatori (oggetto che utilizza il programma per visualizzare dei risultati) con i quali è possibile "costruire" uno strumento hardware virtuale, mentre nel diagramma a blocchi è chiamata **Palette delle Funzioni** e contiene tutte le funzioni e comandi con i quali è possibile "costruire" il programma.

Uno strumento importante è la **Palette degli Strumenti** che contiene tutti gli strumenti con i quali è possibile interagire con un programma LV. Questa palette permette di scegliere strumenti mediante i quali è possibile premere pulsanti, selezionare e spostare oggetti, inserire commenti, tracciare i fili di collegamento tra le funzioni e associare i terminali di una subVI ai controlli e indicatori, di impostare e rimuovere dei breakpoint per il debugging (è possibile verificare i dati in ogni punto del diagramma a blocchi) e di impostare il colore di primo piano e di sfondo di un oggetto, di un pannello ecc... Inoltre, vi è la possibilità di utilizzare la palette degli strumenti in modalità 'Automatic Tool Selection' mediante la quale è possibile la selezione automatica dello strumento più adatto all'operazione in corso. Questa modalità non è

consigliabile in progetti complessi in quanto le scelte dello strumento talvolta sono diverse dalle intenzioni dell'utente.

Ogni funzione LV è rappresentata come un'icona dotata di terminali di ingresso e di uscita, alla quale è possibile collegare altre funzioni, costanti, controlli e indicatori. Come già detto, il collegamento avviene tramite fili che, a seconda dello spessore, rappresenta la **dimensione dei dati**: scalare (fili sottili), vettore (fili spessi) e matrice (fili doppi). Inoltre il loro colore identifica la **rappresentazione dei dati**, ad esempio: arancione (numeri reali), blu (numeri interi), verde (variabili booleane) etc...

Come in tutti i linguaggi di programmazione, vi sono i **Comandi Strutturati**: For Loop, While Loop, Case Structure, Sequence ecc... che sono rappresentati come contenitori al cui interno sono inseriti i comandi. I primi tre, come concetto, sono analoghi a quelli presenti in tutti i linguaggi di programmazione, mentre il Sequence permette di eseguire il codice in una sequenza stabilita dal programmatore. Questo in alcune circostanze è fondamentale perché, come detto, LV esegue il codice in parallelo e questo non permette la conoscenza esatta dell'ordine di esecuzione delle operazioni. Con il While Loop e il For Loop sono disponibili degli elementi di memoria chiamati **Shift Register**, con i quali è possibile tenere memoria degli ultimi valori assunti da un parametro (è possibile inserire più Shift Register per avere una memoria a più passi).

Ad ogni controllo e indicatore può essere associata una **Variabile Locale** che permette di leggere o scrivere dentro un controllo o indicatore. Ogni variabile locale è visibile solo all'interno della VI in cui è stata creata. Stessa funzionalità è quella di una **Variabile Globale**, con la differenza che quest'ultima è visibile da tutte le subVI del programma. In progetti complessi, per semplificare la struttura del diagramma a blocchi, si utilizza la **SubVI** che è una VI utilizzata all'interno di altri programmi. Per il passaggio di parametri alla SubVI, a ogni controllo e indicatore può essere associato un terminale. All'interno di un programma LV,

qualora sia necessario, è possibile inserire uno script (ovvero una sequenza di comandi) **Matlab**, in modo da sfruttare le molte routine disponibili. Questo è possibile attraverso l'utilizzo di un emulatore di Matlab, chiamato **MathScript**, che non necessita che Matlab sia installato sul sistema. LV permette anche la **chiamata di librerie dinamiche**.

Come visto LV è molto versatile, infatti, è possibile:

- Realizzare interfacce utente in modo molto semplice e rapido.
- Realizzare codice efficiente grazie alle numerose funzioni predefinite che operano su array e che sono implementate a basso livello.
- Utilizzare script Matlab mediante MathScript.
- Interfacciarsi con qualunque routine di tipo libreria dinamica, scritte con qualsiasi linguaggio di programmazione, così da sfruttare anche i programmi già disponibili.
- Richiamare funzioni di sistema operativo, mediante chiamate a librerie dinamiche di sistema.

CAPITOLO 2: DRIVER NI-DAQmx

2.1 INTRODUZIONE

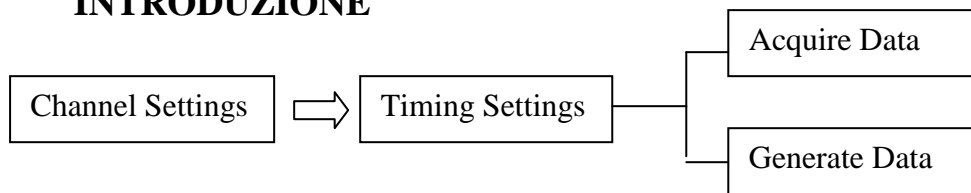


Fig. 2.1: diagramma logico delle funzioni DAQmx utilizzate per l'acquisizione e generazione del segnale.

Per far interagire l'ambiente di sviluppo con il sistema PXI, è stato installato il software NI-DAQmx che è un driver, cioè un insieme di procedure, che tra tutte le sue funzionalità, permette anche di gestire l'acquisizione/trasmissione dati. Come rappresentato nella figura 2.1, i passi logici fondamentali per l'acquisizione e generazione del segnale sono:

- Channel Settings: crea i canali fisici e virtuali da utilizzare.
- Timing Settings: imposta i tempi per l'acquisizione e/o generazione del segnale, come ad esempio la frequenza di campionamento e il numero di campioni da acquisire o generare.
- Acquire Data o Generate Data: permettono di leggere o scrivere i dati su un task.

Questo driver, consente di programmare la scheda *NI PXIe-6361* con linguaggi di alto livello (come ad es. LabVIEW). Di seguito ne saranno descritte le caratteristiche, le potenzialità e le *funzioni DAQmx* che sono state utilizzate nel progetto.

2.2 NI-DAQmx

NI-DAQmx include funzioni avanzate per lo sviluppo dell'acquisizione dei dati e delle applicazioni di controllo. Controlla ogni aspetto del sistema DAQ (Data Acquisition) (inclusi i dispositivi con condizionamento del segnale), la configurazione e la programmazione mediante LabVIEW, il sistema operativo di basso livello e il controllo dei dispositivi. Comprende MAX (Measurement & Automation Explorer) per la configurazione dei dispositivi, test panel per il

debug delle misure, diagrammi di connessione per il collegamento di segnali e sensori e un wizard di calibrazione canali per calibrazione end-to-end da sensore a software.

Mentre DAQ Assistant offre un metodo rapido e veloce per acquisire o generare i dati minimizzando la programmazione, per operazioni più avanzate potrebbe essere necessario un controllo a livello più basso rispetto a quello che è in grado di offrire DAQ Assistant. Per questo, NI-DAQmx driver include una API (Application Programming Interface) completa con funzioni di base e avanzate per il controllo dei parametri di temporizzazione, sincronizzazione, manipolazione dati e controllo di esecuzione. Queste funzioni sono le stesse per tutte le famiglie di dispositivi e per le varie funzionalità che ogni dispositivo offre. Ciò significa che tutte le schede NI-PXIe per acquisizione/generazione dati, sono programmate con lo stesso set di funzioni, quindi una implementazione utilizzata per la gestione di una scheda può essere utilizzata per tutte; inoltre, alcune funzioni sono uguali sia per l'acquisizione che per la generazione. Ad esempio programmare il dispositivo per acquisire un segnale esterno richiede le stesse funzioni di temporizzazione, sincronizzazione e controllo di esecuzione necessarie a generare dei segnali digitali in uscita.

Una caratteristica importante è la **gestione della sincronizzazione** che di solito è un processo di difficile implementazione perché segnali di trigger e/o clock devono essere gestiti manualmente, invece NI-DAQmx esegue automaticamente segnali di routing tra le diverse aree funzionanti di un singolo dispositivo e fra più dispositivi. Un altro requisito dell'architettura NI-DAQmx è il **multithreading**; infatti, più operazioni di acquisizioni dati possono avvenire simultaneamente, migliorando significativamente le prestazioni. L'aspetto pratico più interessante è che risulta sufficiente gestire pochissime funzioni di base per sviluppare un vasto numero di applicazioni di misura. DAQmx è disponibile per diversi linguaggi di programmazione, sia grafici come LabView sia testuali come il C/C++ e Visual Basic: in questa tesi vengono analizzate solo le funzioni utilizzate in LabView.

2.3 CANALI FISICI, CANALI VIRTUALI E TASKS

Per capire bene le funzioni DAQmx, bisogna prima introdurre tre concetti base: i canali fisici, i canali virtuali e i tasks.

2.3.1 Canale fisico

Il canale fisico è un terminale o un pin del dispositivo in uso sul quale è possibile eseguire una misura, ad esempio generare o acquisire un segnale. A seconda del tipo di applicazione, un canale fisico può essere composto da uno o più terminali. Ogni canale fisico, possiede un nome univoco che rispetta una determinata convenzione. Secondo tale convenzione, il nome di un canale fisico consiste *nell'identificatore del dispositivo* e da uno slash (/), seguiti *dall'identificatore del canale*. L'identificatore del dispositivo viene generato in automatico dal sistema e di default è DevN, dove N è un numero progressivo assegnato in base all'ordine con cui i vari dispositivi vengono installati, ad esempio Dev0 e Dev1. L'identificatore del canale combina la tipologia del canale, come "analog input" (ai), "analog output" (ao) e counter (ctr) con un numero intero, ad esempio ai2, ao0, ctr0. Per i canali digitali può essere specificata una porta, per esempio port0, che comprende tutte le linee disponibili.

2.3.2 Canale virtuale

Un canale virtuale è una convenzione dell'architettura del driver NI-DAQmx utilizzata per rappresentare un insieme di proprietà delle configurazioni del dispositivo che può includere un nome, un canale fisico, connessioni del terminale d'ingresso, il tipo di misura o informazioni di generazione e scalatura. Queste informazioni, sebbene non siano strettamente necessarie da un punto di vista fisico per eseguire una misura, sono molto utili in quanto consentono al sistema non solo di formattare correttamente i dati ottenuti, ma anche di ottimizzare l'hardware interno per l'operazione che si vuole eseguire. Quindi un canale virtuale è un software che incapsula un canale fisico.

2.3.3 Tasks

Rappresenta concettualmente un'operazione che si vuole effettuare, sia essa una misura o la generazione di un segnale. Dal punto di vista software, un task è un insieme di uno o più canali virtuali con l'aggiunta di alcune proprietà, ad esempio la temporizzazione o il triggering dei segnali. Tutti i canali virtuali appartenenti ad uno stesso task devono essere della stessa tipologia, cioè o di ingresso o di uscita. La gestione dei task in molte applicazioni avviene in maniera quasi automatica ed è invisibile al programmatore, in quanto il software è in grado di creare e distruggere un task in modo del tutto indipendente. Ad esempio, quando si crea un canale virtuale senza specificare un task di appartenenza, viene creato automaticamente un nuovo task senza ulteriori notifiche all'utente. Il task appena creato verrà poi distrutto nel momento in cui il sistema riterrà opportuno farlo. In genere quando viene completata un'operazione di acquisizione o generazione di un segnale. Questo meccanismo ha tuttavia dei limiti, che riducono notevolmente l'efficienza di un'applicazione in alcuni casi particolari. Inoltre può risultare necessario intervenire sulla gestione di un task manualmente, ad esempio per obbligare il software a validare i parametri impostati dall'utente tramite un eventuale pannello frontale. Per soddisfare questo genere di esigenze, la libreria mette a disposizione delle particolari funzioni avanzate, che permettono di modificare lo stato di un task e quindi di controllarne l'esecuzione.

2.4 FUNZIONI DAQmx

Come accennato, un modo facile e veloce per acquisire e/o generare dati è l'utilizzo del *DAQ Assistant*, che è una interfaccia grafica per la creazione, la modifica e l'esecuzione di canali virtuali e tasks NI-DAQmx.



Fig. 2.2: DAQ Assistant

Alcune applicazioni, però, richiedono più flessibilità e prestazioni maggiori rispetto a quelle fornite dal *DAQ assistant*. Queste applicazioni richiedono l'uso delle *funzioni NI-DAQmx*. Come accennato, queste funzioni permettono lo sviluppo di una vasta gamma di applicazioni, in cui ogni funzione si distingue dalle altre in base al tipo e al numero di parametri in ingresso. Dato che ci sono svariate funzioni ed alcune di esse presentano diverse istanze (corrispondono al tipo specifico di misura o di generazione del canale virtuale) a cui sono associati diversi ingressi, di seguito verranno descritte solo quelle che sono state utilizzate per lo sviluppo del progetto.

2.4.1 NI- DAQmx Flatten Channel String



Fig. 2.3: funzione Flatten Channel String

Questa funzione, rappresentata in figura 2.3, converte i nomi di un array di canali fisici o virtuali in una lista di nomi, e questo permette di usare contemporaneamente più uscite in modo da visualizzare più segnali insieme.

2.4.2 NI-DAQmx Create Virtual Channel

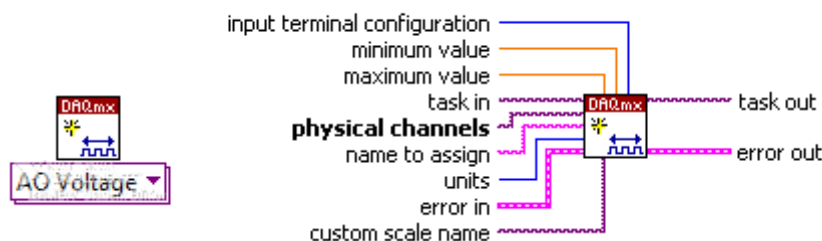


Fig. 2.4: funzione Create Virtual Channel. In particolare sono mostrati l'istanza Analog Output (AO) e i possibili parametri d'ingresso.

La funzione, rappresentata in figura 2.4, è sicuramente una tra le più utilizzate quando si programma con la libreria DAQmx, in quanto serve a creare, come suggerisce il nome, un nuovo canale virtuale da assegnare ad un task. Se non viene specificato alcun task, la libreria ne crea uno in automatico, assegnandogli il canale virtuale.

Nel progetto in esame, è stata usata la configurazione *AO Voltage* per generare un segnale analogico in tensione. I parametri d'ingresso utilizzati sono stati:

- minimum value, maximum value: indicano la minima e la massima tensione che si vuole generare.
- Physical channel: indica i canali fisici da utilizzare in questo canale virtuale

2.4.3 NI-DAQmx Timing

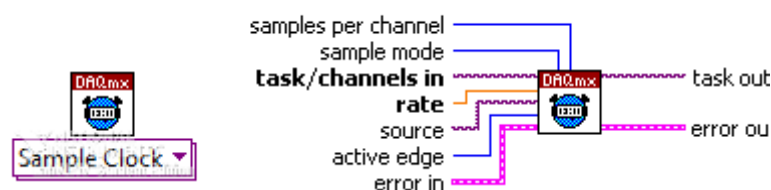


Fig. 2.5: funzione Timing. In particolare sono mostrati l'istanza Sample Clock e i possibili parametri d'ingresso.

Il blocco "Timing VI", rappresentato in figura 2.5, configura i tempi per Hardware-Timed che effettuano operazioni di acquisizioni dati. Sia nel caso dell'acquisizione che in quello della generazione di campioni, non inserendo il blocco "Timing VI" si ottiene il funzionamento "on demand", in cui i campioni vengono acquisiti o generati immediatamente e senza alcuna temporizzazione, ad esclusione dei limiti fisici del dispositivo come, ad esempio, la massima frequenza di campionamento. Per implementare un segnale di clock al fine di regolare l'acquisizione di campioni analogici, occorre specificare "Sample Clock". I parametri d'ingresso utilizzati sono stati:

- **sample mode:** indica la modalità di campionamento da eseguire. Si è utilizzata la modalità "Continuous Samples", in cui vengono acquisiti campioni fino a quando il task si trova nello stato "running".
- **rate:** indica la frequenza di campionamento (sampling rate), espressa come numero di campioni da acquisire al secondo.
- **source:** specifica il terminale da utilizzare come sorgente del clock, nel caso si voglia utilizzare un clock esterno. Nel progetto si è scelto di utilizzare il clock interno al dispositivo.
- **task-channels in:** specifica il nome del task

2.4.4 NI-DAQmx Write

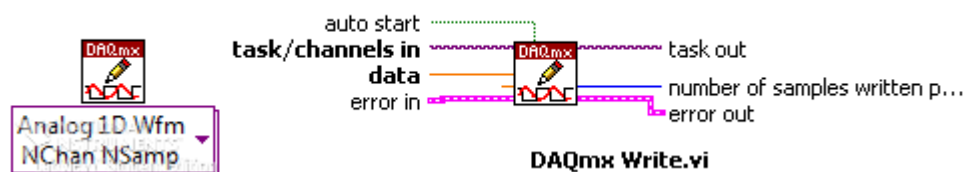


Fig. 2.6: funzione Write. In particolare sono mostrati l'istanza Analog 1D Wfm NChan NSamp e i possibili parametri d'ingresso.

Il blocco "Write VI", rappresentato in figura 2.6, scrive campioni per il task di generazione specificato. La configurazione usata nel programma, scrive uno o più waveform (forme d'onda) in un task che contiene uno o più canali di uscite analogiche. I parametri utilizzati sono stati:

- **data:** contiene i campioni da scrivere
- **task-channels in:** specifica il nome del task

2.4.5 NI-DAQmx Read



Fig. 2.7: funzione Read. In particolare sono mostrati l'istanza e i possibili parametri d'ingresso.

La funzione "Read VI", rappresentata in figura 2.7, serve per leggere dei campioni da un task o da uno o più canali virtuali. La seguente configurazione permette di leggere più forme d'onda da un canale analogico, in particolare (NChan) permette di leggere da più canali virtuali e (NSamp) di acquisire più campioni.

2.4.6 NI-DAQmx Start Task



Fig. 2.8: funzione Start Task

Il blocco "Start Task", rappresentato in figura 2.8, permette di portare manualmente un task nello stato "running", alterando dunque la gestione automatica dei tasks della libreria DAQmx.

2.4.7 NI-DAQmx Clear Task



Fig. 2.9: funzione Clear Task

La funzione, rappresentata in figura 2.9, cancella il task specificato. Se il task è in esecuzione, la funzione prima ferma il task e poi rilascia tutte le risorse fisiche e virtuali da esso occupate. Una volta che un task è stato cancellato, non può più essere utilizzato (si deve ricreare). Quindi se un task dovrà essere riutilizzato, la funzione da utilizzare sarà *NI-DAQmx Stop Task* e non *NI-DAQmx Clear Task*.

2.4.8 NI-DAQmx Properties



Fig. 2.10: DAQmx Property Nodes

Le DAQmx Property, forniscono accesso a tutte le proprietà associate ad una operazione di acquisizione dati. Molte proprietà, come detto, possono essere impostate usando le funzioni NI-DAQmx. Tuttavia, alcune delle proprietà meno frequentemente usate possono essere applicate solo attraverso le NI-DAQmx Property. Queste proprietà possono essere impostate sia per la scrittura che per la lettura. Nel caso di LabVIEW, figura 2.10, vengono chiamate DAQmx Property Nodes.

CAPITOLO 3: SISTEMA DI TRASMISSIONE

3.1 Introduzione

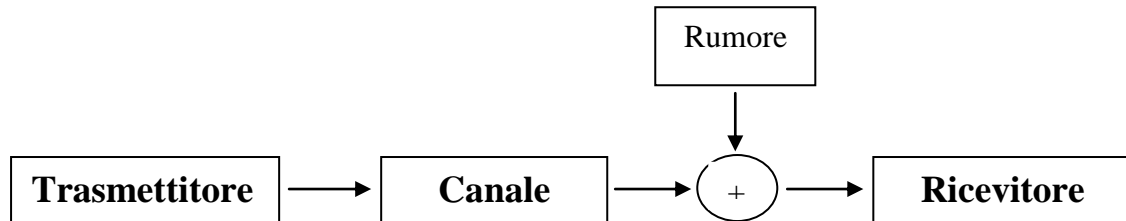


Fig. 3.1: modello sistema di comunicazione

Un sistema di telecomunicazione è un'infrastruttura che serve per implementare una comunicazione tra utenti. Come mostrato in figura 3.1, le parti essenziali di un sistema di comunicazione sono:

- *Trasmettitore*, prende il segnale informativo analogico o digitale e lo converte in un segnale adatto ad essere trasmesso sul canale di comunicazione disponibile.
- *Canale fisico*, è il mezzo trasmissivo (spazio libero, cavo coassiale, fibra ottica, acqua, etc.) utilizzato per la trasmissione del segnale. In alcuni casi si fa uso del termine *canale di comunicazione* per indicare la *porzione* (spettrale o temporale), di un certo canale fisico.
- *Ricevitore*, ha il compito di estrarre dal segnale ricevuto il messaggio trasmesso che a causa del canale e del rumore viene degradato.

Ovviamente, a seconda del tipo di sistema di comunicazione che si vuole realizzare e del canale, i blocchi trasmettitore e ricevitore saranno differenti; in particolare a partire da uno schema di base, verranno aggiunti altri apparati e introdotte nuove tecniche di elaborazione del segnale, in modo da ottenere le massime prestazioni. Per questo motivo, inizialmente è stato realizzato un sistema di comunicazione digitale classico, che utilizza modulazioni M-QAM o M-PSK e, a

partire da questo, sono stati aggiunti nuovi blocchi e tecniche per simulare il funzionamento di un *modem acustico*.

3.2 Sistema di comunicazione digitale generico

I blocchi principali di un sistema di comunicazione digitale generico sono rappresentati in figura 3.2. Uno dei concetti importanti in un sistema di comunicazione (analogico e digitale) è la modulazione. Con questo termine si intende il processo di mappatura dell'informazione, generata da una sorgente, in un segnale che è consono alla trasmissione attraverso un canale fisico. Nel caso delle trasmissioni digitali l'informazione è costituita da una sequenza di bit $\underline{b} = [b_0, b_1, \dots, b_l]$ con $b_i \in (0,1)$, $i = 0,1, \dots, l$ generati con cadenza T_b . La trasformazione della sequenza \underline{b} in un segnale $S_{Tx}(t)$ adatto alla trasmissione sul canale è definita modulazione digitale. Analogamente l'operazione inversa di ricostruzione della sequenza $\hat{\underline{b}}$, dal segnale $r(t)$ ricevuto dal canale, è definita demodulazione (con modulazione/demodulazione si intende l'operazione di mapping e sagomatura del segnale a partire dai bit da trasmettere, e viceversa). In pratica si stabilisce un numero fisso n di bit che compongono le sequenze b_l e si applica una bit-map cioè una relazione biunivoca tra tutte le possibili $Z = 2^n$ combinazioni di n bit e un set di livelli predefiniti ' a_i ' chiamati simboli, appartenenti ad un opportuno alfabeto A , costituito da $Z = 2^n$ elementi.

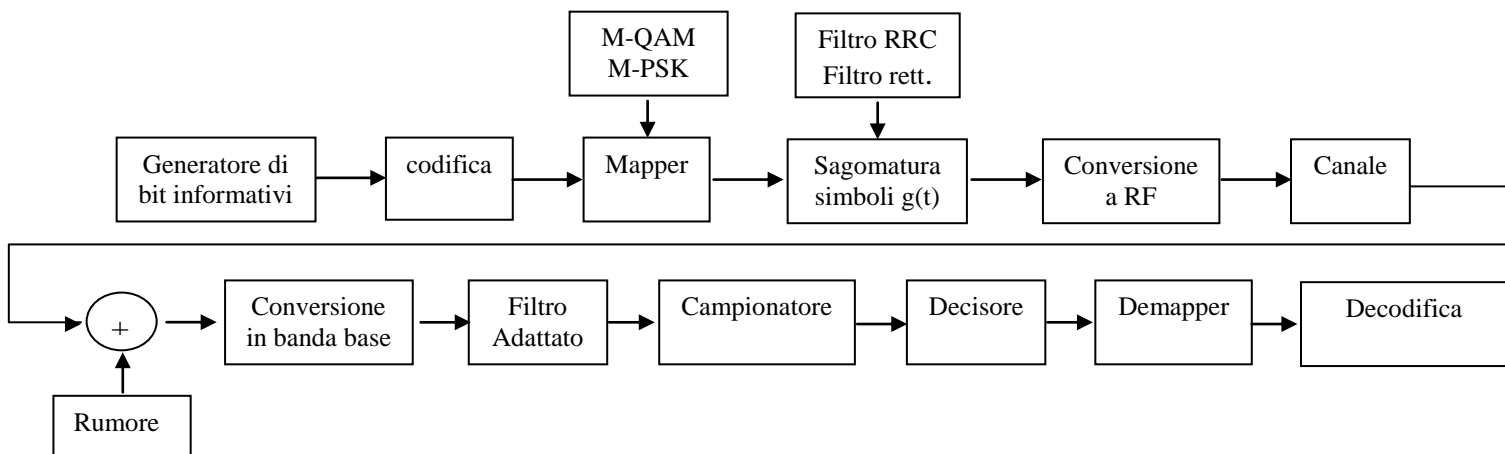


Fig. 3.2: schema a blocchi sistema di trasmissione generico

Una prerogativa della rappresentazione in formato digitale dell'informazione è la codifica di canale che rappresenta uno dei vantaggi delle trasmissioni digitali rispetto a quelle analogiche. Il termine **codifica di canale** designa, in una trasmissione digitale, l'insieme di tecniche di elaborazione del segnale informativo volte a garantire il più possibile la trasmissione ottimale di un messaggio, ovvero l'integrità dei dati attraverso un canale rumoroso che introduce errori nella trasmissione dei dati. In particolare si tratta di tecniche che introducono *ridondanza* in un flusso informativo digitale utilizzato poi per la rilevazione e/o la correzione dell'errore a valle in ricezione. La sequenza di bit in uscita al codificatore deve essere mappata in una sequenza di simboli. Le tecniche di modulazione utilizzate sono svariate e tutte consistono nel variare singolarmente o congiuntamente, ampiezza, fase e frequenza degli impulsi (M-PSK, M-QAM, M-FSK, ecc.). Successivamente, questa sequenza di simboli viene convertita in un segnale adatto ad essere inviato sul canale trasmissivo, tramite un filtro "sagomatore" con una determinata risposta impulsiva $g_T(t)$ (i segnali effettivamente trasmessi sul canale di comunicazione sono di tipo analogico) tale per cui la convoluzione del segnale ricevuto con il filtro di ricezione $g_R(t)$, dà luogo ad un segnale in cui è possibile ridurre alcuni effetti indesiderati introdotti dal sistema di comunicazione e cioè interferenza intersimbolica (ISI) e rumore. Prima di inviare il segnale sul canale può risultare necessaria una conversione in banda-passante e questo dipende dal canale fisico a disposizione (esistono canali fisici passa-banda e canali fisici passa-basso).

Il segnale in uscita dal trasmettitore transita attraverso il canale di telecomunicazione. Questo è un elemento logico e fisico indispensabile all'interno di un sistema di telecomunicazioni, in quanto come mezzo di trasmissione permette il passaggio dell'informazione da una sorgente ad uno o più destinatari. Un canale di trasmissione *ideale* dovrebbe possedere una banda sufficientemente larga ed uniforme, sia in ampiezza che in fase, per contenere lo spettro del segnale di informazione senza distorcerlo; inoltre dovrebbe poterlo trasferire a qualunque

distanza senza introdurre attenuazioni. Nei canali *reali* tipicamente sono invece presenti fattori di degradazione quali: attenuazione, distorsione in banda e rumore; inoltre, il segnale trasmesso, a causa delle strutture presenti nell'ambiente, è soggetto a fenomeni di riflessione, rifrazione e scattering. Esso, quindi, raggiunge il ricevitore per mezzo di cammini multipli (*multipath*), in cui ogni percorso sarà caratterizzato da un valore aleatorio di ritardo, attenuazione e sfasamento. La risposta impulsiva di un canale multipath è:

$$c(t) = \sum_{u=0}^{U-1} h_u \delta(t - \tau_u)$$

Con:

- U = Numero cammini;
- h_u = attenuazione complessa n-esimo raggio
- τ_u = ritardo cammino n-esimo rispetto al raggio diretto (si assume che il raggio diretto abbia ritardo nullo $\tau_0 = 0$)

La combinazione di tutti i percorsi, al ricevitore, produrrà una versione distorta del segnale trasmesso, e quindi presenza di ISI.

Visto che il canale non è un elemento ideale, in ricezione, dopo aver effettuato la conversione in banda-base, si ha tutta una serie di blocchi che servono per il recupero del segnale utile:

- *Filtro Adattato*, come citato sopra, permette l'annullamento (o quasi) di ISI e la riduzione di rumore. Si dice **adattato**, perché la sua risposta impulsiva deve essere uguale, a meno di un coefficiente, a quella del segnale ricevuto. Per una trattazione più completa fare riferimento al capitolo **Sagomatura in banda-base**.
- *Campionatore*, una volta che il segnale esce dal filtro adattato si procede ad effettuare una operazione di campionamento con cadenza T (T = intervallo di segnalazione del simbolo) ottenendo un valore $z_k = a_k + n_k$, dove a_k è il simbolo trasmesso e n_k è un campione di un processo di rumore gaussiano a media nulla e con varianza $N_0/2$.

- *Decisore*, il campione, come detto, è corrotto da rumore e quindi si rende necessaria una regola di decisione per stimare il simbolo trasmesso. Questa regola di decisione consiste nello stabilire delle zone di decisione ottime, che variano a seconda della modulazione utilizzata, in modo da ottenere la minima probabilità di errore.
- *Demappatore*, operazione inversa del mappatore. Dalla sequenza di simboli stimati si ottiene una sequenza di bit stimati.

3.3 Modem Acustico

Lo schema di principio a livello fisico del modem acustico è riportato in figura 3.3, in cui sono evidenziati i principali blocchi funzionali della catena di trasmissione. Il sistema ha caratteristiche simili a quelle del modem Evologics S2CR 18/34.

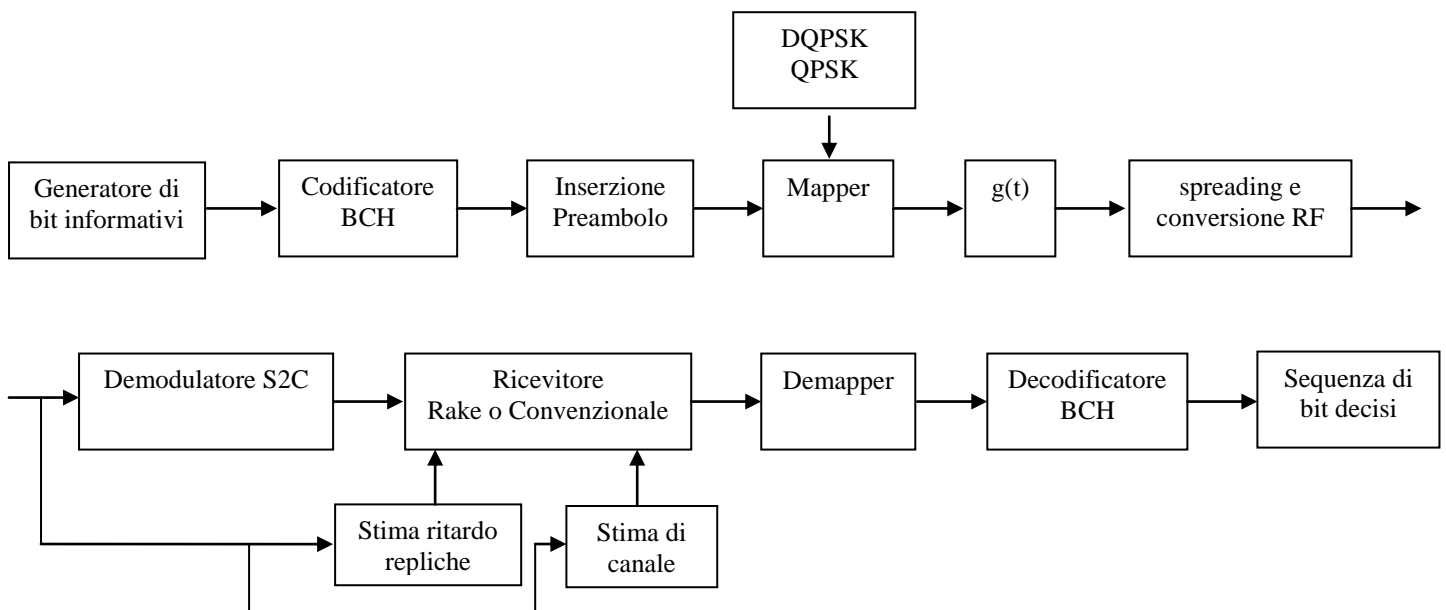


Fig. 3.3: schema di principio a livello fisico del Modem

3.3.1 Codifica a protezione d'errore

Il flusso di bit di informazione è protetto contro gli errori di decodifica dovuti al canale di propagazione a mezzo di decodifica FEC (Forward Error Correction). La codifica di canale utilizzata dal modem proposto si basa sull'utilizzo dei codici lineari a blocco BCH. Indicata con k la lunghezza del blocco di bit informativi $\mathbf{b} = [b_0, b_1, \dots, b_{k-1}]$ in ingresso al codificatore e con n la lunghezza della corrispondente parola codificata $\mathbf{c} = [c_0, c_1, \dots, c_{n-1}]$, si definisce *rate* del codice il rapporto $r = k/n$ con $0 < r < 1$.

I codici BCH appartengono alla categoria dei codici ciclici, in cui la generazione della parola codificata e la successiva decodifica si basano sulle proprietà algebriche dei polinomi definiti nei campi di Galois (GF). Ogni codice BCH è identificato univocamente da un polinomio generatore e la lunghezza della parola codificata è pari a $n = 2^m - 1$ con m intero, mentre risulta $k \geq n - mt$, dove t è il numero massimo di errori per parola di codice che è possibile rivelare e correggere, in quanto la distanza minima tra due parole di codice è maggiore o uguale a $d_{\min} = 2t - 1$. Ad esempio, il codice BCH {1431,2047} con $m = 11$ e $t = 58$ è in grado di generare parole di codice con distanza di Hamming (numero di posizioni in cui i due vettori binari differiscono) non inferiore a $d_{\min} = 115$. Le numerose combinazioni $\{k, n, t\}$ ammesse, consentono la generazione di codici con differenti gradi di ridondanza anche fissando la lunghezza della sequenza di uscita n . Nei modem Evologics sono implementati 48 differenti codici con *rate* r variabile, in modo da consentire un buon numero di gradi di libertà al sistema di trasmissione, che è in grado di adattare ed ottimizzare i propri parametri in relazione alle condizioni istantanee di canale.

3.3.2 Mappatura con codifica differenziale

Il vettore di bit codificati \mathbf{c} è suddiviso in gruppi di M bit, ad ognuno dei quali viene associato un simbolo di canale complesso appartenente ad una costellazione di 2^M simboli, ottenendo in uscita il vettore di simboli di canale \mathbf{a} . Nel modem in esame è utilizzata una mappatura PSK

differenziale con $L = 2$ (DQPSK). In questo caso, stante la codifica differenziale, ad ogni possibile coppia di bit è associata una differente rotazione di fase rispetto alla fase trasmessa nel precedente intervallo di segnalazione. In questo modo, impiegando un rivelatore differenziale ed accettando una modesta degradazione delle prestazioni (dell'ordine di un paio di dB) rispetto ad una modulazione PSK coerente, è possibile rendere il sistema robusto nei confronti di eventuali offset di fase e di frequenza al ricevitore, in quanto l'informazione è codificata nella differenza di fase tra intervalli di segnalazione consecutivi e non nella fase assoluta.

3.3.3 Sagomatura in banda base

Una volta ottenuti i simboli complessi, si procede alla sagomatura in banda base degli stessi inviando la parte reale e immaginaria rispettivamente al filtro sagomatore del ramo in fase (I) ed al filtro sagomatore del ramo in quadratura (Q) del trasmettitore. Di seguito, senza perdita di generalità, si supporrà di operare, come è convenzione, con simboli e segnali complessi senza considerare la distinzione tra i rami I e Q. Dopo il filtro sagomatore si ha quindi il segnale elettrico con espressione:

$$s(t) = \sum_i a_i g_T(t - iT)$$

dove T è l'intervallo di segnalazione (ovvero il tempo che trascorre tra l'invio di due simboli consecutivi) e $g(t)$ è la risposta impulsiva del filtro di trasmissione. Una soluzione progettuale molto comune è quella di scegliere la risposta impulsiva complessiva dei filtri di trasmissione e ricezione $g_T(t) \otimes g_R(t)$ in modo tale da approssimare un impulso di Nyquist, vale a dire, idoneo a produrre in uscita un segnale che, dopo il campionamento, con timing perfetto a cadenza $1/T$ e senza rumore, restituisca esattamente la sequenza di simboli trasmessi senza alcuna interferenza intersimbolica. Una famiglia di impulsi di Nyquist molto utilizzati è quella con sagomatura a coseno rialzato, ottenuta imponendo che le trasformate di Fourier di $g_T(t)$ e $g_R(t)$ abbiano andamenti a radice di coseno rialzato (RCR). Nel dominio del tempo ciò implica:

$$g_T(t) = g_R(t) = \begin{cases} 1 - \alpha - \frac{4\alpha}{\pi} & \text{per } t = 0 \\ \frac{\alpha}{\sqrt{2}} \left[\left(1 + \frac{2}{\pi}\right) \sin\left(\frac{\pi}{4\alpha}\right) + \left(1 - \frac{2}{\pi}\right) \cos\left(\frac{\pi}{4\alpha}\right) \right] & \text{per } t = \pm \frac{T}{4\alpha} \\ \frac{\sin\left(\pi \frac{t}{T}(1 - \alpha)\right) + 4\alpha \frac{t}{T} \cos\left(\pi \frac{t}{T}(1 + \alpha)\right)}{\pi \frac{t}{T} \left[1 - \left(4\alpha \frac{t}{T}\right)^2\right]} & \text{altrove} \end{cases}$$

dove α è il fattore di roll-off e il suo valore (compreso tra 0 e 1) indica l'eccesso di banda dell'impulso rispetto alla funzione a banda minima ($1/T$ in banda passante) che soddisfa la condizione di Nyquist. Valori tipici di α sono intorno a 0.2-0.25. Nelle implementazioni reali dei filtri di trasmissione e ricezione la sintesi delle risposte impulsive, teoricamente di durata infinita, avviene limitandone l'estensione a qualche intervallo di segnalazione.

3.3.4 Espansione spettrale

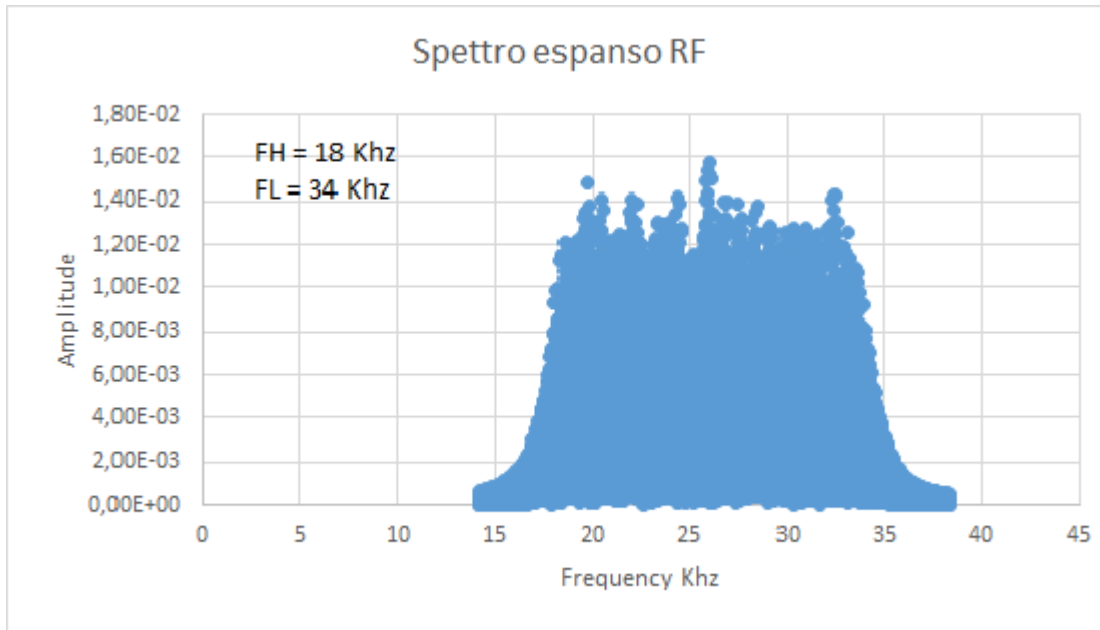


Fig. 3.4: segnale a spettro espanso inviato sul canale

Una volta che il segnale è stato generato in banda base esso deve essere convertito alla frequenza di trasmissione sul canale acustico (radiofrequenza) attraverso il blocco di modulazione, rappresentato in figura 3.3 dal blocco "spreading e conversione a RF". La

particolarità della tecnica di modulazione utilizzata nel modem in esame, derivata dai modem Evologics e nota come **Sweep-Spread Carrier** (S2C), consiste nell'impiego di una portante che, oltre a traslare in alto la banda del segnale, provoca anche una espansione spettrale dello stesso, figura 3.4. Il segnale a radiofrequenza dopo conversione in alto ed espansione è dato da:

$$x(t) = \Re\{s(t) * c(t)\}$$

dove $c(t)$ è un fasore rotante a frequenza non costante, del tipo

$$c(t) = \exp \left[j2\pi \left(f_L \left(t - \left\lfloor \frac{t}{T_{sw}} \right\rfloor T_{sw} \right) + m \left(t - \left\lfloor \frac{t}{T_{sw}} \right\rfloor T_{sw} \right)^2 \right) \right]$$

dove $m = \frac{f_H - f_L}{2T_{sw}}$ con f_H e f_L pari rispettivamente ai limiti di banda superiore ed inferiore. Questi due limiti di frequenza, insieme *all'intervallo di sweep* T_{sw} definiscono la pendenza $2m$ della rampa di frequenza e caratterizzano (come risulterà più chiaro in seguito) la capacità di risoluzione dei percorsi multipli presenti nel canale da parte del ricevitore. La frequenza istantanea della portante è data infatti dalla derivata dell'argomento interno dell'esponenziale che definisce $c(t)$ all'interno di un intervallo di sweep, ovvero:

$$f_i(t) = f_L + 2m \left(t - \left\lfloor \frac{t}{T_{sw}} \right\rfloor T_{sw} \right)$$

il cui andamento in funzione del tempo è rappresentato, in termini di pulsazione, in figura 3.5

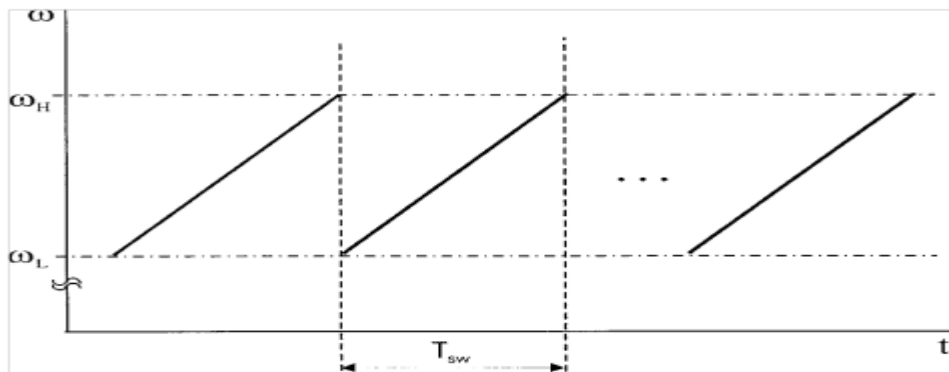


Fig. 3.5: Frequenza istantanea del segnale portante.

3.3.5 Despreading

Dal punto di vista spettrale, dopo la modulazione S2C, il segnale occupa una banda all'incirca compresa tra f_L e f_H che risulta quindi espansa di un fattore $M = \frac{f_H - f_L}{\frac{1+\alpha}{T}}$ rispetto ad un segnale con modulazione convenzionale a banda stretta. Per comprendere l'utilità di questo tipo di modulazione in uno scenario di propagazione a cammini multipli è utile modellare il segnale ricevuto $y(t)$ come la sovrapposizione di repliche ritardate ed attenuate del segnale trasmesso, alle quali si somma un contributo di rumore¹:

$$y(t) = \sum_{u=0}^{U-1} h_u x(t - \tau_u) + w(t)$$

$$x(t) = s(t) * c(t)$$

dove si può ritenere che il ritardo relativo al primo cammino sia nullo ($\tau_0 = 0$), considerando gli altri come ritardi relativi. Si ammette inoltre che il ricevitore sia in grado di sincronizzare esattamente la portante locale con quella relativa alla componente di segnale che arriva sul cammino più forte, (di norma quello diretto). L'espressione del segnale dopo l'operazione di demodulazione e ricompattamento spettrale (despreading) è quindi data da:

$$z(t) = y(t)c^*(t)$$

Questa operazione trasla la componente di segnale ricevuta sul cammino diretto a cavallo della continua; essa può pertanto essere recuperata attraverso un semplice filtro passa-basso. A titolo di esempio si supponga $U = 3$, primo raggio con ritardo nullo e assenza di rumore. In questo caso si ha:

$$z(t) = h_0 s(t) + h_1 s(t - \tau_1) c^*(t) c(t - \tau_1) + h_2 s(t - \tau_2) c^*(t) c(t - \tau_2)$$

$$|c(t)|^2 = 1$$

¹Si suppone che durante l'intervallo di osservazione il canale rimanga statico, omettendo quindi la dipendenza dal tempo di U , h_u e τ_u

dove il primo termine a secondo membro può essere isolato a mezzo di un filtro passa-basso, senza subire distorsioni da parte delle altre repliche del segnale, a condizione che queste ultime siano sufficientemente lontane dalla prima nel dominio della frequenza, figura 3.6.

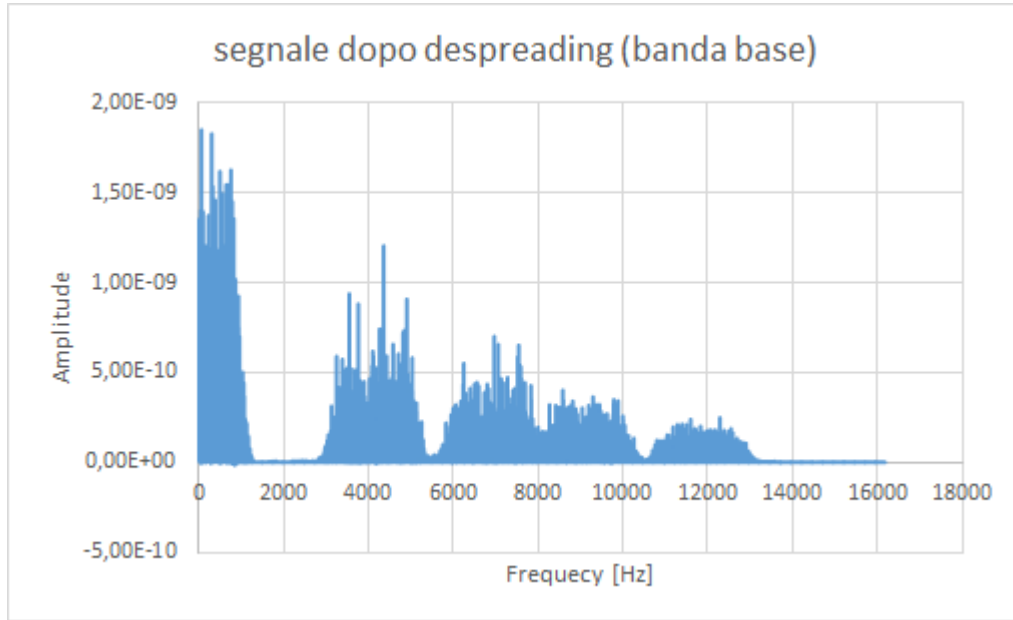


Fig. 3.6: spettro in frequenza dopo l'operazione di despreading ottenuto utilizzando un canale a tre raggi.

Sviluppando il prodotto $c^*(t)c(t - \tau_1)$ si ottiene un segnale con frequenza istantanea, entro l'intervallo $(0, T_{sw})$, pari a:

$$f_i^{(1)}(t) = \begin{cases} -2m\tau_1 + f_H - f_L & 0 \leq t < \tau_1 \\ -2m\tau_1 & \tau_1 \leq t < T_{sw} \end{cases}$$

dove si è supposto $\tau_1 < T_{sw}$. In altri termini, la replica del segnale $s(t - \tau_1)$ viene traslata in frequenza, rispetto alla componente ricevuta sul primo raggio (riportata sulla continua), della quantità $-2m\tau_1 + f_H - f_L = (f_H - f_L)(1 - \frac{\tau_1}{T_{sw}})$ per un primo tratto della rampa utilizzata nel despreader, mentre nella parte rimanente della rampa la traslazione di frequenza è pari a $-(f_H - f_L) \frac{\tau_1}{T_{sw}}$.

Allo stesso modo, la replica del segnale ricevuta sul terzo raggio viene traslata sulla frequenza $(f_H - f_L)(1 - \frac{\tau_2}{T_{sw}})$ nell'intervallo di tempo $(0, \tau_2)$ e su $-(f_H - f_L)\frac{\tau_2}{T_{sw}}$ in (τ_2, T_{sw}) . Riassumendo, quindi, il segnale associato al primo raggio è riportato esattamente sulla continua, mentre le altre due repliche danno luogo, in frequenza, a segnali a banda stretta traslati rispetto al primo.

Da questo esempio risulta chiaro come la pendenza della rampa (ovvero il dimensionamento di T_{sw} una volta fissati i limiti f_L e f_H) sia determinante per risolvere, cioè riuscire a separare, i vari cammini. Infatti, le condizioni da soddisfare simultaneamente per non far ricadere repliche indesiderate del segnale all'interno della banda passante del filtro passa-basso, sono le seguenti:

$$2m\tau_{min} > \frac{1 + \alpha}{T} \rightarrow T_{sw} < \frac{(f_H - f_L)T\tau_{min}}{1 + \alpha} = M\tau_{min}$$

$$-2m\tau_{max} + f_H - f_L > \frac{1 + \alpha}{T} \rightarrow T_{sw} > \frac{(f_H - f_L)T\tau_{max}}{T(f_H - f_L) - (1 + \alpha)} = \frac{M}{M - 1}\tau_{max}$$

dove $\tau_{max} = \max\{\tau_0, \tau_1, \dots, \tau_{U-1}\}$ e $\tau_{min} = \min\{\tau_0, \tau_1, \dots, \tau_{U-1}\}$. L'ultima relazione diviene $T_{sw} > \tau_{max}$ se $M \gg 1$. Si trovano quindi delle soluzioni al problema suddetto a condizione che

$$1 < \frac{\tau_{max}}{\tau_{min}} < M - 1$$

Nella discussione precedente si è ipotizzato che il ricevitore effettui il despreading in modo da sintonizzarsi sulla replica meno ritardata ($u = 0$). Si segnala che il modem Evologics prevede una procedura che consente periodicamente di identificare, fra le repliche di segnale ricevute, quella a maggior contenuto energetico. Una volta ottenuta la stima di timing ottima per questa replica, l'algoritmo di despreading procede come nell'esempio precedente, andando a moltiplicare l'intero segnale ricevuto per un segnale portante generato localmente e sincronizzato con tale replica. Non sempre la replica a massima potenza è quella a ritardo minimo, ed in tal caso alcuni dei ritardi differenziali tra il cammino più forte e gli altri cammini sono negativi. Ciò non altera comunque la validità

generale dei risultati e delle condizioni ottenute con la discussione precedente, purché si consideri per τ_{max} e τ_{min} rispettivamente il valore massimo e quello minimo dei valori assoluti dei ritardi differenziali.

Dopo despreading e filtraggio passa-basso, se la struttura multipath del canale è risolvibile, si ottiene un segnale a banda stretta privo di interferenza cui è sommato rumore. Si procede quindi alla rivelazione del segnale informativo in maniera convenzionale, con filtraggio adattato, stima di canale, campionamento, decisione e decodifica.

3.3.6 Stima del ritardo

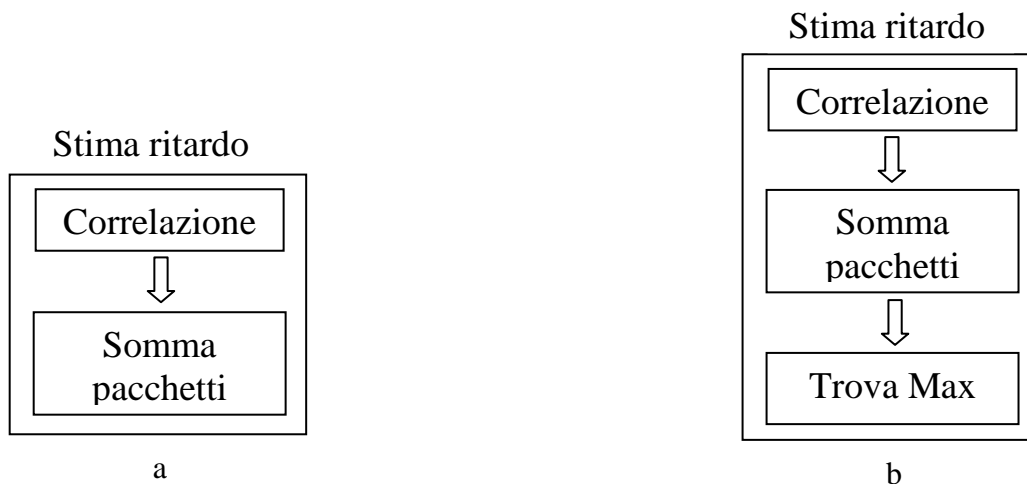


Fig. 3.7 : a) ricevitore a singolo raggio; b) ricevitore RAKE

Una funzione importante nel ricevitore è rivestita dal metodo di stima del ritardo con cui giungono in ingresso al ricevitore i segnali presenti sui vari cammini. Una tecnica convenzionale adottata nel caso di **ricevitore a singolo raggio**, figura 3.7 a, è quella di stimare solamente il ritardo del segnale a maggior contenuto energetico. Se invece si vogliono ottenere prestazioni migliori, bisogna sfruttare il massimo numero possibile² di repliche utilizzando un **ricevitore a RAKE**, figura 3.6 b, che, però, comporta la stima dei

²Il numero massimo dipende da quante repliche soddisfano alcune condizioni e dal numero di correlatori presenti nel ricevitore a RAKE.

ritardi di tutte le repliche. Quindi, se da una parte si ottengono prestazioni migliori dall'altra si aumenta la complessità del sistema.

Come rappresentato in figura 3.7, le operazioni utilizzate per una efficiente stima del ritardo sono:

- Correlazione
- Somma valori di correlazione
- Trova massimo

che a seconda del ricevitore utilizzato, possono essere implementate tutte o in parte. Nel caso di ricevitore a singolo raggio, è sufficiente trovare il valore massimo del segnale dopo l'operazione di correlazione (basta fare un ordinamento decrescente del vettore e prendere il primo valore), mentre nel caso di ricevitore a RAKE è stata sviluppata la funzione **"Trova massimo"**, che, come descritto in seguito, stima i ritardi di tutte le repliche che soddisfano determinate condizioni. Infine, la funzione **"Somma valori di correlazione"**, che può essere utilizzata sia nel caso di ricevitore a singolo raggio che a RAKE, serve per rendere il sistema più efficiente nel caso di valori elevati di rumore. Di seguito, dapprima viene descritto come implementare l'operazione di correlazione per la stima del raggio a maggior contenuto energetico e, successivamente, vengono esposte le procedure con le quali sono state realizzate le altre 2 funzioni e i problemi che risolvono.

3.3.6.1 stima del raggio a maggiore potenza

Per agevolare l'operazione di stima, in ogni pacchetto dati è presente un preambolo costituito da N_s intervalli di durata T_{sw} (corrispondente a N_s rampe complete di frequenza portante), all'interno dei quali sono trasmessi dei simboli noti, utilizzati anche per la successiva stima dei coefficienti h_0, \dots, h_{U-1} del canale. Senza perdita di generalità si suppone noto al ricevitore l'istante di arrivo del primo raggio, che è possibile facendo

ricorso ad un rivelatore d'energia. La stima del ritardo di arrivo della replica più forte viene effettuata a mezzo di tecniche di correlazione, come segue:

$$\hat{\tau} = \arg \max_{\tilde{\tau} \in \{0, T_{sw}\}} \int_{\tilde{\tau}}^{\tilde{\tau} + N_s T_{sw}} y(t - \tilde{\tau}) s_T^*(t) dt$$

andando cioè a cercare l'istante in cui l'ultimo tratto di segnale ricevuto, di durata $N_s T_{sw}$, è maggiormente "somigliante" al preambolo $s_T(t)$ memorizzato localmente nel ricevitore. Per quanto detto nella sezione precedente, la ricerca del massimo è effettuata per valori di $\tilde{\tau}$ non superiori a T_{sw} . Riportando su un grafico l'espressione sopra riportata in funzione dei valori di tentativo si possono osservare in figura 3.8 dei massimi locali in corrispondenza dei ritardi associati alle repliche ricevute e tra questi il massimo globale identifica la replica principale.

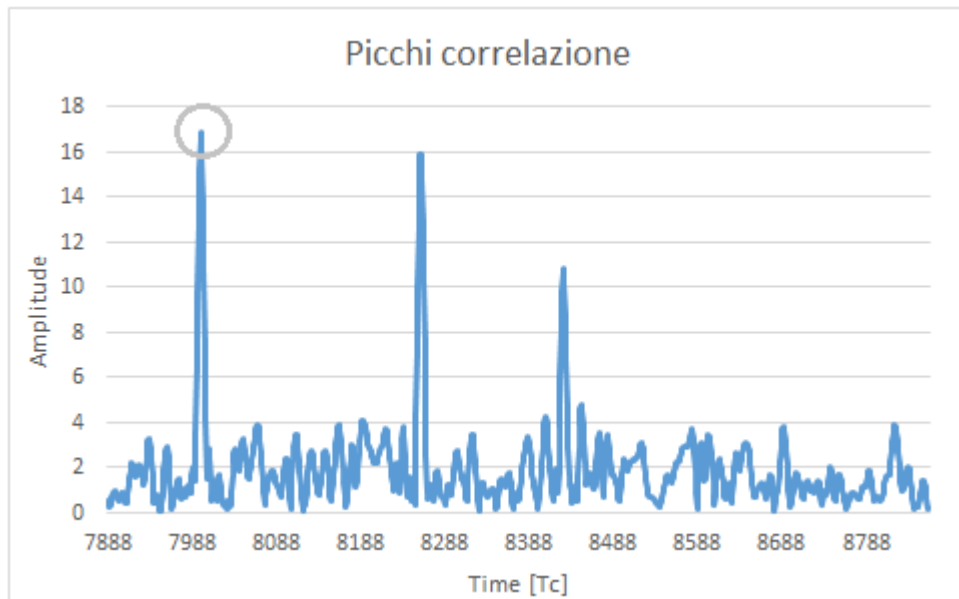


Fig. 3.8 : segnale dopo l'operazione di correlazione

3.3.6.2 Stima di più raggi

Come visto, un canale a cammini multipli può essere modellato mediante un filtro che ha una risposta all'impulso:

$$c(t) = \sum_{l=1}^L h_l \delta(t - \tau_l)$$

questo fa sì che al ricevitore arrivino numerose copie del segnale, ciascuna delle quali presenta un ritardo differente dipendente dalla diversità spaziale dei cammini.

Come descritto nel paragrafo precedente, un ricevitore convenzionale può essere progettato in modo che il despreader si sincronizzi sul cammino a maggiore potenza (cioè quello con $|h_n|$ più elevato). Se, invece, si vogliono ottenere prestazioni migliori si possono combinare tali segnali utilizzando il **ricevitore RAKE** che consente di ottenere un guadagno dalla diversità dovuta ai cammini multipli. In sostanza è possibile costruire un ricevitore dotato di più despreader, ognuno sincronizzato su un cammino diverso. Ad esempio, il j -esimo despreader ($j = 1, \dots, L$) moltiplica il segnale ricevuto per la sequenza $c(t - \hat{\tau}_j)$ sincronizzandosi sul raggio con ritardo τ_j ; i segnali in uscita da ogni despreader vengono elaborati utilizzando metodi convenzionali (filtro adattato e campionatore) e infine i campioni vengono combinati assieme usando, ad esempio, la tecnica **MRC** (*maximal ratio combining*) che sarà descritta in seguito. Come esempio, in figura 3.9 è rappresentato un canale ad N raggi con differenti ritardi e differenti attenuazioni, ognuno dei quali corrisponde ad un differente cammino di propagazione fra quelli possibili. Il ricevitore Rake ha un blocco di ricezione per tre differenti cammini multipli ed in ognuno di questi viene fatto il despreading, cioè, il segnale ricevuto viene correlato con il codice di spreading complesso coniugato e ritardato della quantità stimata, in modo da allinearsi temporalmente con la replica, dovuta al multipath, del segnale ricevuto.

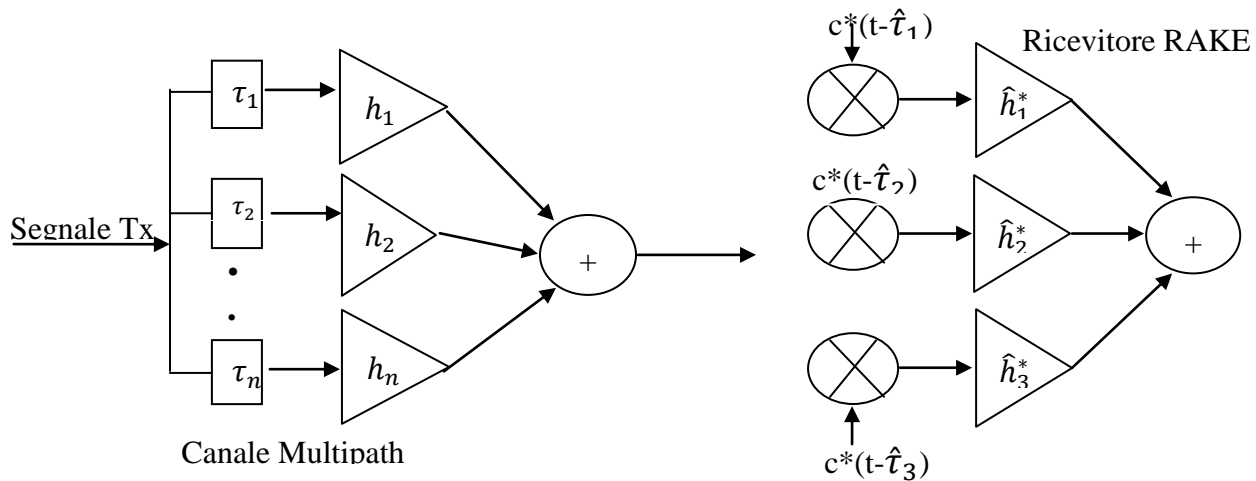


Fig. 3.9: canale multipath e ricevitore RAKE

3.3.6.3 Trova massimi

Come detto precedentemente, vi sono dei massimi locali in corrispondenza dei ritardi associati alle repliche ricevute, che contengono informazione. Da questa considerazione è nata l'idea di sfruttare non solo il raggio a maggiore potenza, ma anche tutti gli altri raggi che possono essere considerati "*utili*" ai fini di un miglioramento delle prestazioni, in termini di bit error rate (BER), del sistema. Quello che ci si prefigge di fare con la funzione **Trova massimi** è proprio quello di selezionare i picchi "*utili*". Questa selezione viene fatta sulla base delle seguenti due condizioni:

1. Tutti i valori di correlazione che stanno al di sotto di una *soglia* non vengono considerati.
2. Fra i valori rimasti si selezionano i massimi che sono distanti almeno un intervallo di scorrelazione.

Inoltre per diminuire la complessità computazionale dell'algoritmo, è stato operato un *ordinamento decrescente indicizzato*, sul segnale di correlazione. Per capire cosa questo significhi, si consideri la tabella 3.1 a , che rappresenta il vettore in ingresso alla funzione di ordinamento. In uscita alla funzione, tabella 3.1 b, si ottiene un vettore ordinato in modo decrescente i cui indici rappresentano la posizione del numero nel vettore non ordinato.

indici	Vettore non ordinato		indici	Vettore ordinato
0	10	→	0	10
1	5		6	9
2	7		3	8
3	8		2	7
4	4		1	5
5	2		4	4
6	9		5	2

Tabella 3.1: a) vettore non ordinato; b) vettore ordinato mantenendo gli indici del vettore non ordinato

Questa operazione di ordinamento serve per ottimizzare il codice.

Le condizioni, come mostrato in figura 3.10, si basano su due parametri: *Soglia* e *Intervallo di scorrelazione*:

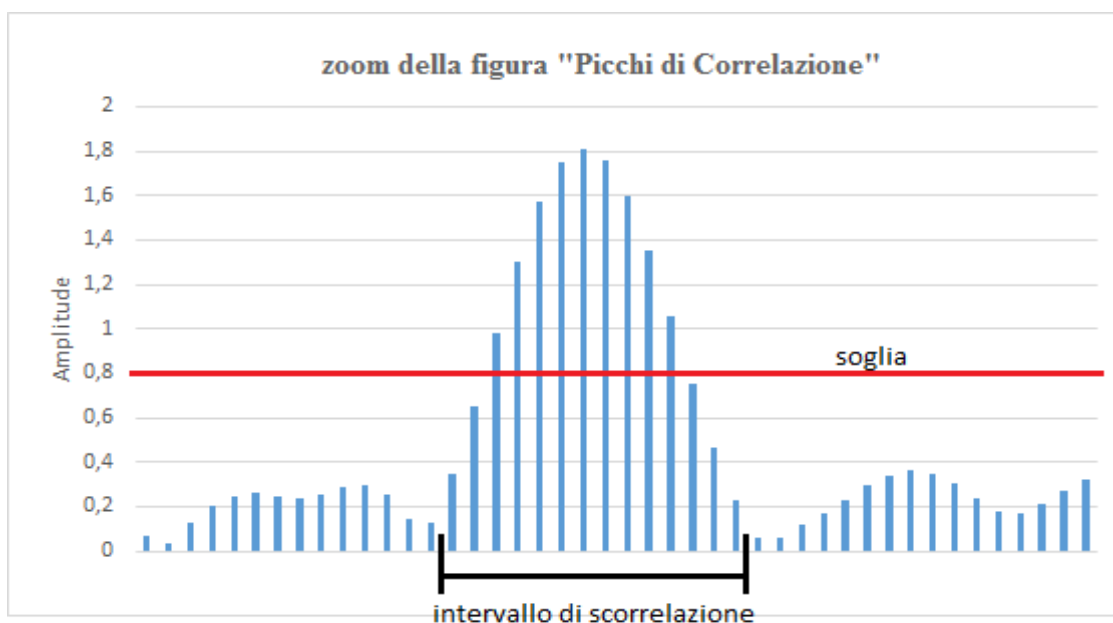


Fig. 3.10 : rappresenta, virtualmente, come la funzione "trova massimi" usa la soglia e l'intervallo di scorrelazione.

1. La soglia permette di selezionare solo quelle repliche che hanno un valore energetico che può essere significativo ai fini di un miglioramento della *BER*. Inoltre, si ha una notevole riduzione del *tempo di calcolo* perché tutte le repliche sotto la soglia vengono scartate. In più, facendo un ordinamento decrescente, non

appena la funzione trova un valore sotto la soglia vuol dire che anche tutti quelli successivi non soddisfano la condizione e quindi possono essere scartati ottimizzando ulteriormente il tempo di calcolo. Come esempio, si consideri che al ricevitore arrivino due raggi (uno diretto e l'altro riflesso) che producono, dopo l'operazione di correlazione, il risultato mostrato in figura 3.11 e di disporre di 3 despreader (si possono sfruttare massimo 3 repliche).

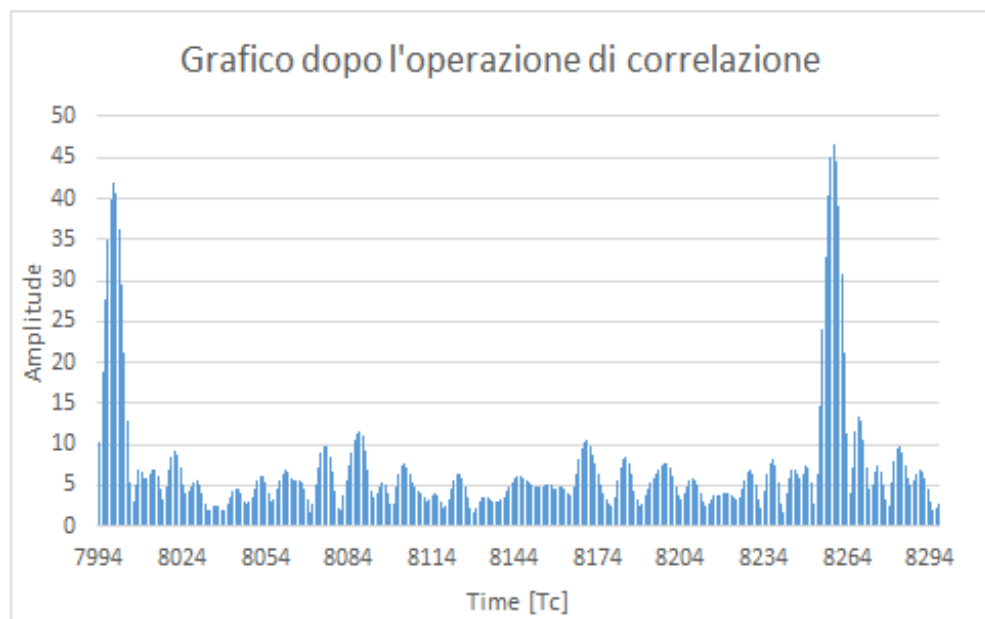


Fig. 3.11: zoom sull'intervallo delle due repliche; grafico ottenuto dopo l'operazione di correlazione

Se la soglia fosse impostata in maniera corretta si otterrebbe la situazione rappresentata in figura 3.12, in cui si hanno i 2 massimi associati ai 2 raggi e il sistema funzionerebbe in maniera ottima.

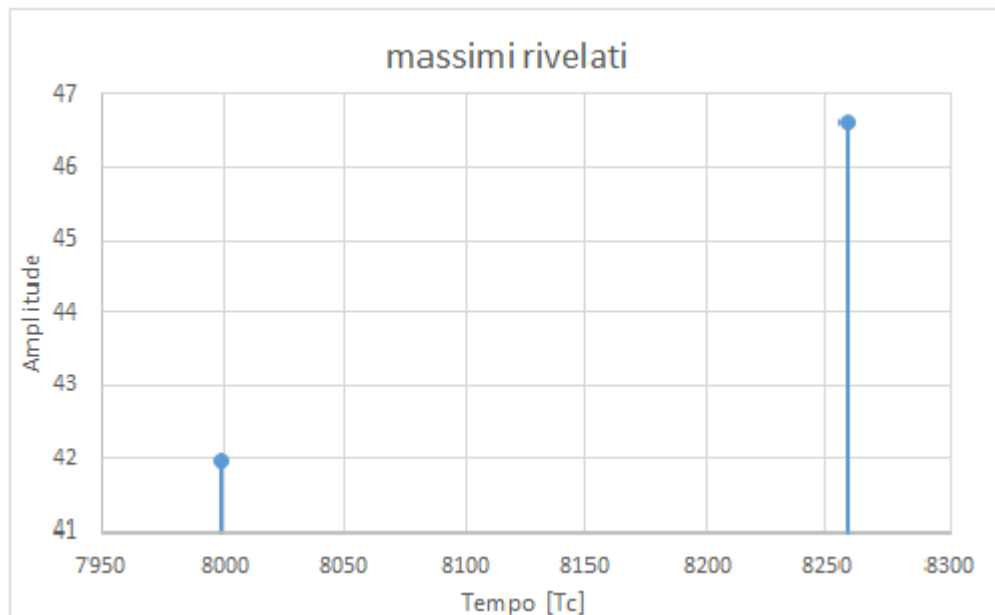


Fig. 3.12: mostra quali sono i massimi trovati dalla funzione "trova massimi".

Diverso sarebbe il caso in cui la soglia fosse settata male; in questa situazione c'è il rischio di rivelare ulteriori massimi, figura 3.13, non dovuti a raggi riflessi del segnale e questo causerebbe l'attivazione del terzo despreader degradando le prestazioni del sistema sia in termini di BER che di tempo di calcolo. Questo esempio ci fa capire come sia importante scegliere la soglia in maniera adeguata in modo da non rivelare picchi di correlazione *spuri* cioè non dovuti ad un reale cammino di propagazione, oppure comunque troppo piccoli per comportare un incremento significativo di prestazioni.

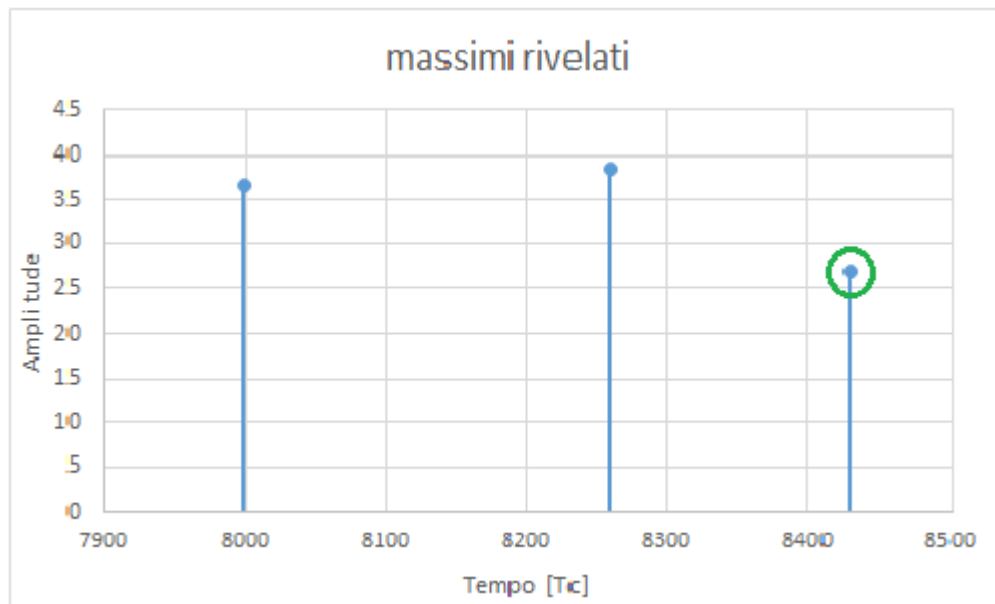


Fig. 3.13: massimi trovati di cui uno (quello cerchiato in verde) non è *utile*

2. L'intervallo di scorrelazione $T_{sc} \cong \frac{1}{B}$ $B = f_H - f_L$ è l'intervallo di tempo in cui il picco di correlazione associato al massimo è poco correlato con un altro valore (in figura 3.10 si vede che i valori agli estremi dell'intervallo di scorrelazione non hanno alcun relazione con il picco massimo). Questo parametro consente di non considerare come riferimenti i picchi che vengono fuori dall'operazione di correlazione. Per capire meglio il concetto, si consideri la figura 3.10 in cui è visualizzato lo zoom del segnale dopo l'operazione di correlazione. La linea orizzontale in rosso indica la *soglia* che, come detto, seleziona i valori che stanno al di sopra di essa. Questa condizione, però, non è sufficiente per determinare correttamente i massimi, perché ci sono dei campioni della correlazione del segnale ricevuto con il preambolo locale, che non sono da considerare picchi *utili*. Per risolvere questo problema, si è introdotto il concetto di *intervallo di scorrelazione*: si inizia andando a cercare il massimo assoluto a cui si associa un range (linee verticali nere in figura 3.10), pari a:

$$range_1 = I_1 \pm kT_{sc}$$

$range_1$ = range associato al primo massimo

I_1 = indice, del vettore ordinato, associato al primo massimo

$k = 1, 2, 3 \dots$ (Vi è anche la possibilità, come visto nell'espressione, di settare il range come multiplo dell'intervallo di scorrelazione, diminuendo il tempo di ricerca dei massimi).

Una volta trovato, si cerca il secondo massimo³ in tutto lo spazio dei campioni tranne che nel range associato al primo massimo. Questa operazione viene fatta andando a vedere se l'indice del vettore non ordinato associato al secondo massimo, cade all'interno del range associato al primo massimo. Se questo non si verifica, il secondo massimo esiste e il range associatogli viene concatenato al range del primo massimo;

$$range_2 = I_2 \pm kT_{sc}$$

$$range_{tot} = range_1 \& range_2 \quad tot = totale$$

La funzione continua andando a cercare il terzo massimo in tutto lo spazio dei campioni tranne che nei due range associati al primo e al secondo massimo ($range_{tot}$), e così via per la ricerca di tutti gli altri massimi. La funzione termina quando elabora tutto il vettore passatogli e cioè il vettore formato da tutti i valori sopra la soglia.

3.3. 6.4 Somma valori di correlazione

Un problema che può verificarsi a bassi rapporti segnale-rumore è la possibilità che il riferimento scelto per la stima del ritardo non sia quello corretto. Questo comporta un

³ I massimi vengono cercati in ordine decrescente.

errore di allineamento e quindi del calcolo di τ con conseguente degrado della BER. Ad esempio, si consideri la figura 3.14, in cui il picco *utile* (freccia gialla) è superato, a causa del troppo rumore, da un picco non *utile* (freccia verde) che essendo quello a maggiore potenza verrà utilizzato dal sistema come riferimento per il calcolo di τ con conseguente errore nell'allineamento della portante (despreader) con il segnale ricevuto. Un modo per aumentare la probabilità di stimare correttamente τ è di non basare il suo calcolo su un singolo pacchetto ma su N pacchetti. Supponendo che in un pacchetto ci siano M campioni, la funzione **somma valori di correlazione** memorizza un numero N di pacchetti che poi va a sommare (si sommano i valori di correlazione situati nella stessa posizione) andando così a formare un nuovo pacchetto C di lunghezza P su cui verrà stimato τ . In termini matematici possiamo scrivere:

$$C_j = \sum_{i=1}^N a_{ji} \quad j = 1 \dots P$$

a_{ji} = campione j -esimo dell' i -esimo pacchetto

Effettuando questa operazione è come se la varianza (N_0) del rumore diminuisse di un fattore N e quindi a parità di E_b/N_0 si ottengono prestazioni migliori.

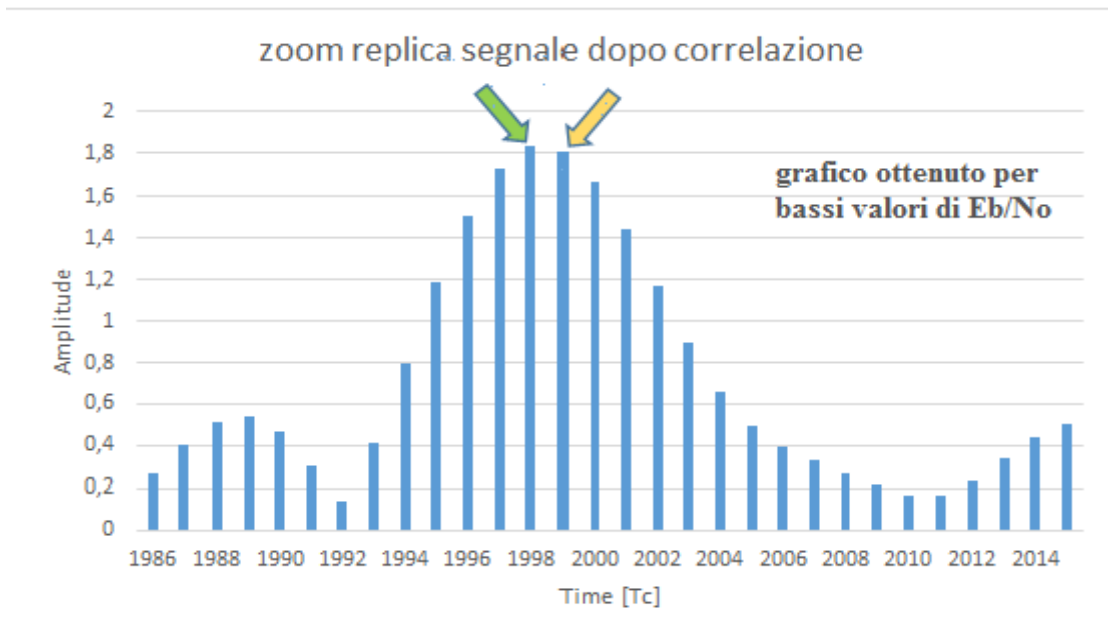


Fig. 3.14: dal grafico si osserva un picco non utile, freccia verde, può avere un maggior contenuto energetico di un picco utile, freccia gialla.

3.3.7 Stima di canale

Lo sviluppo di un ricevitore ottimo ha lo scopo di minimizzare la probabilità d'errore in presenza di rumore e di ISI. Il criterio di Nyquist specifica quali sono le condizioni cui deve soddisfare l'impulso ricevuto per non avere ISI. Nei casi reali queste condizioni non sono soddisfatte perchè il segnale ricevuto è il prodotto (in frequenza) del segnale inviato e della risposta in frequenza del canale distortente. Quindi per poter ricostruire il segnale trasmesso in maniera esatta bisogna conoscere con precisione la risposta (nel tempo o in frequenza) del canale. Una conoscenza esatta della risposta del canale non è possibile in quanto il canale introduce come minimo rumore. Per questo ci sono varie tecniche che effettuano la stima di canale. Se la stima di canale è buona, il canale può essere equalizzato cioè possono essere compensate le distorsioni introdotte dal canale quali attenuazione e sfasamento (un'errata stima del canale non consentirebbe una corretta equalizzazione). È necessario fornire all'equalizzatore una stima del canale attraverso il quale è transitato il segnale trasmesso. Quindi, la stima di canale è un'operazione importante perché stima i

coefficienti \hat{h}_n che permettono l'utilizzo di tecniche per migliorare la decisione sul simbolo, come ad esempio l' MRC.

Nel progetto del modem realizzato, la stima di canale è stata ottenuta andando a calcolare modulo (opportunamente normalizzato) e fase dei picchi massimi *utili* ottenuti dopo l'operazione di correlazione del segnale ricevuto con il preambolo locale (la posizione dei picchi massimi è data dalla funzione trova massimi). Così facendo, si vedono quali sono stati gli effetti del canale sui diversi percorsi e cioè l'attenuazione e lo sfasamento che esso ha introdotto. Una volta stimato il canale, i coefficienti calcolati vengono passati all'equalizzatore che ne fa il complesso coniugato, ottenendo \hat{h}_n^* , per poi moltiplicarli per il segnale ricevuto in modo da equalizzare il canale.



Fig. 3.15: procedura utilizzata per ottenere i coefficienti \hat{h}_n

3.3.1.8 Maximal Ratio Combining

Come visto, l'espressione in ingresso al ricevitore nell'ipotesi di multipath, rumore AWGN e canale statico è:

$$y(t) = \sum_{u=0}^{U-1} h_u x(t - \tau_u) + w(t)$$

Questa espressione indica che il canale introduce rumore e che ogni cammino introduce una attenuazione complessa. Come detto, ogni raggio contiene informazioni utili e quindi si è pensato di utilizzare un ricevitore RAKE che è in grado di recuperare un numero massimo di repliche pari al suo numero di rami; tramite le operazione di despreading e filtro passa basso, utilizzate su ogni ramo, si riescono a separare le repliche associate ad ogni cammino. Quindi, disponiamo di più repliche e si deve trovare un metodo efficiente su

come combinarle per prendere la decisione sul simbolo. L'espressione del segnale dopo l'operazione di despreading e filtro passa-basso è:

$$z_n(t) = h_n s(t - \tau_n) + w(t) \quad n = 1 \dots N \quad N = \text{numero di despreader}$$

in cui i coefficienti $h_n = \rho_n e^{j\vartheta_n}$ sono dei valori complessi che corrispondono all'attenuazione e allo sfasamento introdotti da ogni percorso. La variabile discreta (dopo il campionatore) ottenuta su ogni ramo è del tipo:

$$X_n(k) = h_n C_n + W_n(k)$$

kT = istante di campionamento

C_n = simbolo sull'ennesimo ramo

W_n = rumore sull'ennesimo ramo

Si indichi con $Y(k)$ la variabile di decisione ottenuta combinando in maniera lineare i campioni di ogni ramo. Quello che si ottiene è:

$$Y(k) = \sum_{n=1}^N \beta_n * X_n(k)$$

Resta da determinare come scegliere i coefficienti β_n .

Alcuni dei metodi proposti in letteratura sono i seguenti:

- *combinazione a guadagno costante*: si considerano i β_n tutti uguali (in questo modo, però, non "peserei" i campioni ai diversi rami del ricevitore RAKE)
- *combinazione a selezione*: si considera l' $X_n(k)$ con $\beta_n(k)$ massimo (si pongono tutti i $\beta_n = 0$ ad eccezione del β_n relativo al ramo che si è scelto)
- l'ultimo metodo che è quello che massimizza il rapporto segnale-rumore è la tecnica *MRC*

La tecnica MRC consiste nell'equalizzare tutte le repliche e cioè compensare lo sfasamento e l'attenuazione introdotti su ogni cammino moltiplicando per \hat{h}_n^* in modo che i campioni

relativi ad ogni ramo possano essere sommati in fase in modo da ottenere il massimo rapporto segnale-rumore. Quindi è importante avere una buona stima del canale.

Nel progetto del modem realizzato, figura 3.16, viene fatta una operazione di equalizzazione sul segnale in ingresso al Ricevitore che significa moltiplicare il segnale per \hat{h}_n^* ; il segnale equalizzato viene moltiplicato per $c^*(t-\hat{\tau}_n)$ (despreading) e fatto passare attraverso un filtro adattato (queste 2 operazioni corrispondono, come visto, a portare il segnale in banda base e a filtrare in modo da recuperare una sola replica) ottenendo le repliche separate ognuna caratterizzata dal proprio ritardo; infine, su ogni ramo viene effettuato il campionamento⁴. I campioni ottenuti vengono sommati ottenendo la variabile di decisione $Y(k)$ che viene inviata in ingresso al decisore.

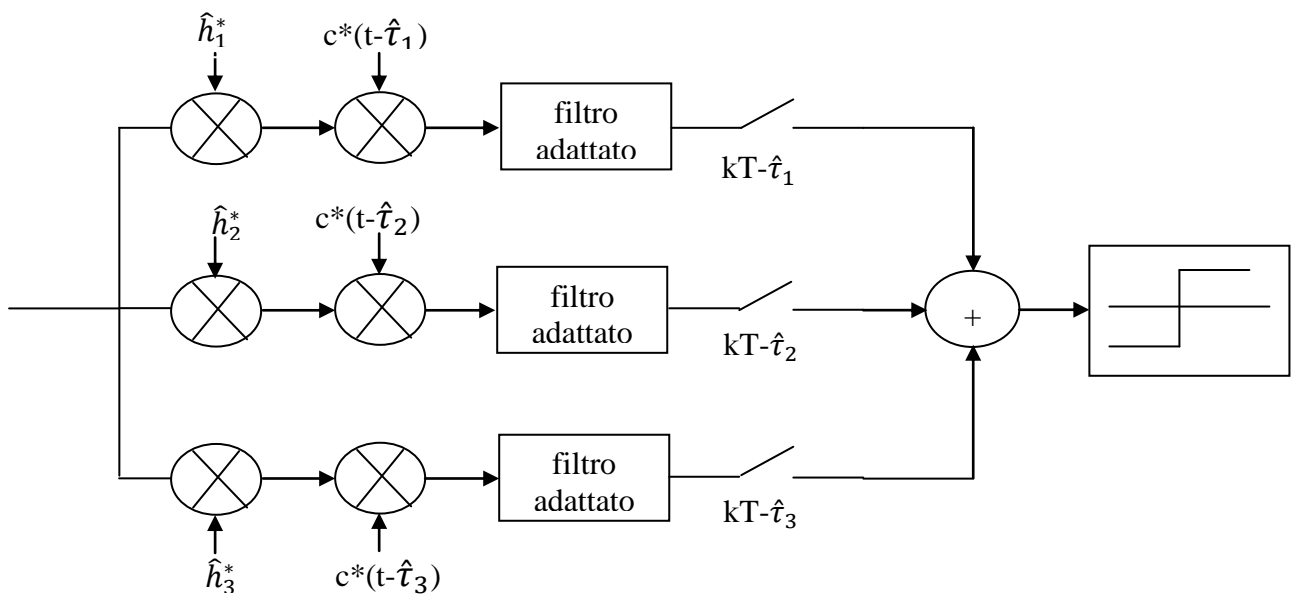


Fig. 3.16: schema di principio di un ricevitore a RAKE in cui si utilizza la tecnica MRC

⁴ Il campionamento su ogni ramo va ritardato di una quantità pari al ritardo introdotto da ogni cammino

CAPITOLO 4: ANALISI SULLE PRESTAZIONI DEL MODEM

4.1 Introduzione

In questo capitolo viene effettuata una analisi sulle prestazioni, in termini di bit error rate (BER), del modem acustico. In particolare verranno valutate le prestazioni del modem in diversi scenari operativi e verranno forniti dettagli su come migliorarne le prestazioni variando alcuni parametri di livello fisico. L'analisi verrà fatta sia utilizzando il ricevitore a singolo raggio che quello a RAKE e dato che le prestazioni dipendono anche dal rapporto E_b/N_0 è utile descrivere come questo parametro è stato trattato.

4.2 Analisi sull'energia e sul rumore

In un qualsiasi sistema di comunicazione che deve trattare informazioni, due dei parametri da tenere in considerazione sono l'energia media per bit informativo ricevuta e il rumore termico. Le prestazioni di un sistema di comunicazione in termini di BER sono tipicamente determinati al variare di E_b/N_0 , che mette in relazione l'energia del segnale utile con quella del rumore. Esso è dunque un parametro di merito molto importante per il dimensionamento ed il corretto funzionamento di un sistema di comunicazione in quanto determina la capacità del ricevitore di rivelare il flusso informativo originario senza incorrere in alterazioni dovute ad errori causati dal rumore. Più basso è questo rapporto, più sarà difficile la decodifica del segnale, ovvero più alta la BER. Un confronto equo tra le varie simulazioni va fatto a parità di E_b/N_0 .

4.2.1 Energia

L'energia considerata per il calcolo della BER è *l'energia media per bit informativo ricevuta*, che dipende da molti fattori: modulazione utilizzata, intervallo di segnalazione del simbolo, filtro sagomatore, codifica e multipath.

Nel sistema realizzato, la modulazione utilizzata è una QPSK con codifica di Gray e la bit-map è la seguente:

Bit	Simboli
00	1+j0
01	0+1j
11	-1+j0
10	0-1j

Tabella 4.1: bit-map modulazione QPSK

Quindi l'energia media per simbolo, che è pari alla distanza quadratica del simbolo dall'origine, è unitaria. Una volta fatta l'associazione bit-simboli, questi vengono sagomati attraverso un filtro con una risposta impulsiva a radice di coseno rialzato (RRC) che è stata realizzata in modo da avere energia unitaria. Una ulteriore considerazione riguarda l'operazione di codifica che, come detto, è una tecnica che introduce *ridondanza* per la rilevazione e/o la correzione dell'errore in ricezione (nel progetto si è utilizzato un codice BCH con k=1431 e n=2047). Infine bisogna tener conto che l'energia che si riceve è quella dovuta a tutte le repliche. Nel progetto, per tenere conto di tutto ciò, viene calcolata la densità spettrale di potenza in ingresso al ricevitore. Il risultato ottenuto viene integrato, trovando la potenza e, per trovare l'energia media per simbolo, si moltiplica per l'intervallo di segnalazione del simbolo $E_s = PT$. Infine, dato che si è interessati all'energia media per bit informativo, l'espressione da considerare è:

$$E_b = \frac{E_s}{\log_2 M * r}$$

M = numero di punti della costellazione

r = k/n rate del codice

4.2.2 Rumore

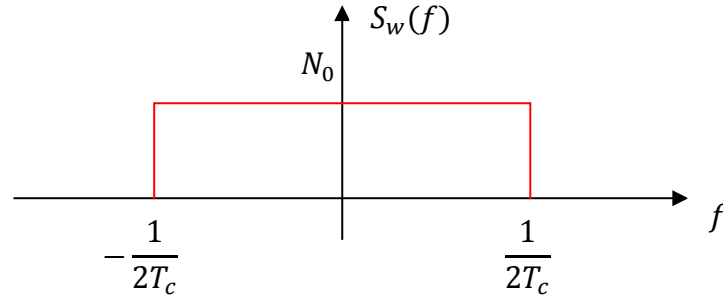


Fig. 4.1: Densità Spettrale di Potenza del Rumore

La qualità di un sistema di comunicazione dipende da quanto il segnale ricevuto riproduca in maniera fedele il segnale trasmesso dalla sorgente. Purtroppo nei sistemi reali, la trasmissione dei segnali è disturbata dalla presenza di *rumore*. Vi sono vari tipi di rumore, ma quello trattato nel progetto è il *rumore termico*. La sua importanza dipende dal fatto che nel progetto di apparati spesso rappresenta il livello minimo di disturbo con il quale il segnale d'informazione deve inevitabilmente convivere.

Nel sistema, si è considerata una *densità spettrale di potenza di rumore* $S_w(f)$ bianca entro un certo range di frequenze $\left(-\frac{1}{2T_c} \div \frac{1}{2T_c}\right)$, figura 4.1. Questo perché il filtro in ricezione limita la banda del segnale e, quindi, l'andamento della $S_w(f)$ a frequenze più elevate della banda del filtro di ricezione non è rilevante (ovvero, all'esterno di tale banda, il rumore può avere la densità spettrale di potenza che si vuole, senza alterare i risultati del sistema). Il vantaggio di scegliere il rumore bianco all'interno del range di frequenze $\left(-\frac{1}{2T_c} \div \frac{1}{2T_c}\right)$ consiste nel fatto che, con tale scelta, i campioni di rumore prelevati a multipli di T_c sono incorrelati tra loro e, in quanto il rumore gaussiano, sono anche indipendenti. Da ciò si deduce che i campioni di rumore possono essere prodotti mediante una funzione che genera campioni gaussiani e indipendenti. Infatti, l'espressione della funzione di autocorrelazione è:

$$R_w(\tau) = TCF[S_w(f)]$$

$$R_w(\tau) = \alpha * \text{sinc}\left(\frac{f}{T_c}\right)$$

TCF = trasformata continua di fourier

α = coefficiente d'ampiezza

questo significa che i campioni di rumore agli istanti kT_c sono incorrelati ed essendo gaussiani sono indipendenti.

4.2.3 Calcolo della BER al variare di E_b/N_0 dB

Per il calcolo della BER è bene fare riferimento al rapporto E_b/N_0 espresso in dB. Il calcolo è stato effettuato per valori crescenti di E_b/N_0 in modo da vedere come variano le prestazioni del modem. In tutte le simulazioni, fissato il valore di E_b/N_0 , si è lasciato inalterato E_b e si è variato N_0 .

4.3 Impostazioni di progetto

Come detto, il modem in esame ha una capacità di auto-adattamento dei propri parametri di trasmissione in relazione alle variazioni delle caratteristiche dell'ambiente di propagazione. I suddetti parametri sono i seguenti:

- T_{sw}
- Lunghezza del preambolo ($N_{sr} = \frac{kT_{sw}}{T}$ lunghezza del preambolo espressa in numero di simboli, k numero intero)
- Roll-off
- Intervallo di segnalazione (T)

Di questi parametri, però, gli unici che sono stati variati nelle diverse simulazioni sono T_{sw} e N_{sr} , mentre gli altri due sono stati mantenuti uguali in tutte le simulazioni:

- Roll-off = 0.3
- T = 0.5 ms

Inoltre, in tutte le simulazioni sono state utilizzate le seguenti impostazioni:

- Codifica BCH (1431, 2047)
- Modulazione QPSK con mappatura di Gray
- Tre pacchetti (indica il numero di pacchetti utilizzati nella funzione **somma valori di correlazione**)
- Frequenza di campionamento (f_c) = 100 kHz (quindi un fattore di sovracampionamento $N_c = f_c T = 50$)
- $f_L = 18000 \text{ Hz}$; $f_H = 34000 \text{ Hz}$

4.4 Simulazioni software

Una volta stabilito come vengono trattati l'energia e il rumore, è possibile effettuare un'analisi sulle prestazioni del modem acustico in diversi scenari. Per fare una buona analisi vengono rappresentati diversi grafici, che sono:

- **Grafico profilo potenza-ritardo:** come spiegato nel capitolo 3, il ricevitore a singolo raggio sfrutta il raggio a maggior contenuto energetico, mentre il ricevitore a Rake elabora fino ad un massimo di tre repliche. Per questo è buona norma rappresentare il grafico **potenza-ritardo** in modo da avere informazioni sulle caratteristiche del canale e valutare di quanto attenuano e ritardano i vari cammini:
 1. potenza normalizzata, indica quanto è il valore di potenza associato ad un cammino rispetto alla potenza complessivamente ricevuta
 2. ritardo relativo, indica dopo quanto tempo arrivano le repliche rispetto alla prima, che si è assunta con ritardo nullo.
- **spettro del segnale dopo despreading:** per capire come l'operazione di spreading e despreading agisce sul segnale multipath, si rappresenta lo spettro del segnale dopo

l'operazione di despreading con sincronizzazione effettuata sul raggio più forte. Si potrebbe rappresentare anche lo spettro del segnale dopo l'operazione di despreading con sincronizzazione sugli altri raggi (non su quello più forte), con la differenza che sulla continua si troverebbe lo spettro della replica sulla quale si è sincronizzati e le altre repliche sarebbero traslate in frequenza, rispetto alla componente sulla continua, di una quantità che dipende dalla differenza di ritardo che tali repliche hanno rispetto alla replica di riferimento.

- **BER:** in ogni sistema di comunicazione è fondamentale misurare questo parametro perché permette di stabilire quali sono le prestazioni del sistema e quindi quale deve essere il minimo valore di E_b/N_o per avere determinate prestazioni.

4.4.1 Primo scenario

Le caratteristiche del canale sono state trovate utilizzando un simulatore di canale acustico subacqueo chiamato *Bellhop*. Questo simulatore per fare una analisi sul canale richiede l'impostazione di alcuni parametri: periodo dell'anno cui si riferisce la simulazione, distanza orizzontale, profondità dell'acqua, quota sorgente e destinatario, diagramma d'irradiazione del proiettore.

In particolare, in questa simulazione l'analisi del canale è stata fatta in base al seguente scenario:

- Inverno⁵
- Numero sorgenti = 1; Numero destinatari = 1
- Profondità acqua 130 m
- Distanza orizzontale tra sorgente e ricevitore pari a 500 m.
- quota di 80 m sia per la sorgente che per il ricevitore

⁵ Fondamentalmente si distinguono 2 periodi: estate e inverno. A seconda del periodo, varia la caratteristica di velocità del suono in funzione della profondità nell'acqua.

- intervallo di generazione raggi rispetto all'orizzonte $-50 \div 6$ gradi

In questo scenario il numero di raggi che sono stati rivelati sono 3 e presentano i seguenti ritardi di propagazione:

- $\tau_{a0} = 0,33092785 \text{ sec.}$
- $\tau_{a1} = 0,34704429 \text{ sec.}$
- $\tau_{a2} = 0,39097875 \text{ sec.}$

τ_{ai} = ritardo di propagazione i-esimo raggio.

Di seguito vengono considerati i ritardi relativi tra i raggi, ipotizzando che il raggio con minore ritardo abbia ritardo nullo.

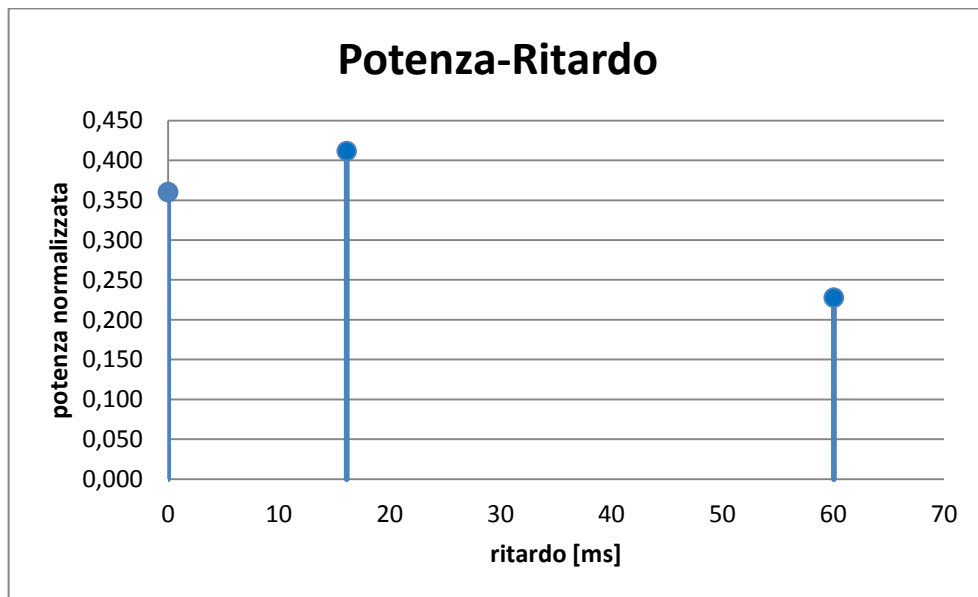


Fig 4.2: rappresenta l'attenuazione e il ritardo introdotti su ogni cammino.

Le caratteristiche del canale sono riportate in figura 4.2, da cui si evince che i ritardi relativi sono:

- $\tau_1 = 16,1 \text{ ms}$
- $\tau_2 = 60,1 \text{ ms}$

Dal grafico si evince che la replica utilizzata dal ricevitore a singolo raggio (quella a contenuto energetico maggiore) è quella con ritardo $\tau_1 = 16 \text{ ms}$ mentre il ricevitore RAKE le sfrutta tutte.

Dall'analisi teorica riportata nel capitolo 3, si è visto che T_{sw} deve rispettare alcune condizioni per far sì che le repliche non siano sovrapposte in frequenza, in modo che possano essere recuperate senza interferire fra di loro. Queste condizioni sono:

$$\bullet \quad T_{sw} < M\tau_{\min} \quad e \quad T_{sw} > \frac{M}{M-1}\tau_{\max} \quad \text{con} \quad M = \frac{(f_H - f_L)T}{1+\alpha}$$

Calcolato il valore di $M = 6.15$ ed essendo $\tau_{\min} = 16 \text{ ms}$ e $\tau_{\max} = 60 \text{ ms}$, si trova che $72 \text{ ms} < T_{sw} < 99 \text{ ms}$ (per semplicità si approssima ad un numero intero). Inoltre N_{sr} deve essere un multiplo di una rampa, quindi multiplo di T_{sw} . Nella simulazione si è scelto $T_{sw} = 80 \text{ ms}$ e $N_{sr} = 160$ (pari alla lunghezza di una rampa). Si consideri che questi valori sono scelti una volta noto il canale; nella realtà non conoscendolo a priori, si deve prevedere un probing preliminare del canale.

Una volta impostato il valore di T_{sw} è utile fare una analisi spettrale del segnale e cioè visualizzare lo spettro del segnale dopo l'operazione di despreading per vedere se effettivamente il risultato ottenuto per simulazione coincide con quello teorico. Prima di fare ciò, è bene ricordare che la banda di un segnale in banda base utilizzando un filtro RRC è pari a $B = \frac{1+\alpha}{2T}$, sostituendo i valori utilizzati nelle simulazioni si ottiene $B = 1300 \text{ Hz}$ (vedi figura 4.3). Come detto, in teoria, si dovrebbe ottenere uno spettro in cui il segnale associato al raggio più forte è riportato esattamente sulla continua, mentre le altre due repliche danno luogo, in frequenza, a segnali a banda stretta, di banda B , traslati rispetto al primo. In particolare, come rappresentato in figura 4.3, la replica più forte è quella con ritardo 16 ms che quindi viene riportata sulla continua, questo significa che le altre 2 repliche (quella con ritardo nullo e quella con ritardo 60 ms) è come se avessero un ritardo, rispetto alla componente più forte, di:

- $\tilde{\tau}_1 = 16 - 0 = 16 \text{ ms}$
- $\tilde{\tau}_2 = 60 - 16 = 44 \text{ ms}$.

Da questi risultati si ricavano, utilizzando la formula del capitolo 3, i seguenti valori di frequenza istantanea:

- $f_i^1(t) = 3200 \text{ Hz}; 12800 \text{ Hz}$
- $f_i^2(t) = 7200 \text{ Hz}; 8800 \text{ Hz}$

che coincidono con quelli trovati per simulazione, figura 4.3, in cui è visualizzato lo spettro relativo alle sole frequenze positive. Dal grafico, ottenuto per simulazione, si evince come sia possibile recuperare la replica con l'utilizzo di un semplice filtro passa basso e questa operazione può essere fatta, se gli spettri sono separati, per tutti i raggi, basta mettere un filtro in uscita da ogni despreader. Il grafico è stato ottenuto in assenza di rumore per una migliore visualizzazione (la presenza del rumore termico non fa spostare le frequenze dello spettro e quindi si otterrebbero gli stessi valori di frequenza istantanea).

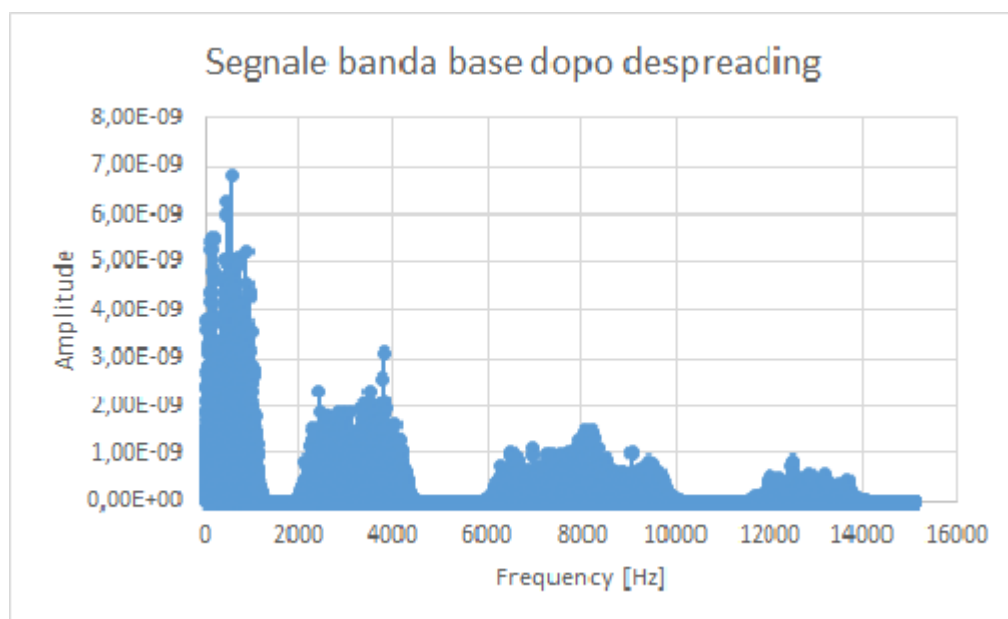


Fig. 4.3: spettro ottenuto dopo il despreading fatto sul raggio più forte.

Una volta fatta l'analisi sul canale e sullo spettro del segnale dopo il despreading, si può rappresentare il grafico relativo alla BER sia nel caso in cui si utilizzi il ricevitore che sfrutta il raggio più forte (tecnica convenzionale) sia utilizzando quello a Rake.

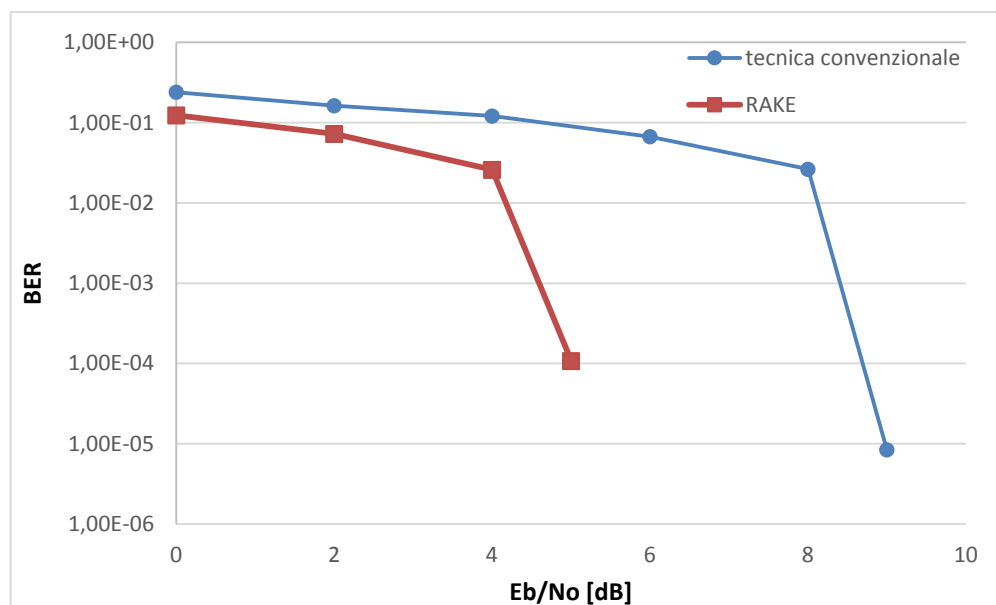


Fig. 4.4: BER per vari valori di E_b/N_0 nel caso di ricevitore a singolo raggio e RAKE

Come si vede in figura 4.4, l'utilizzo del ricevitore a Rake comporta delle prestazioni migliori in quanto sfrutta l'informazione contenuta in tutte e tre le repliche. Inoltre, in entrambi i casi, anche se a differenti valori di E_b/N_0 , la pendenza della curva cambia in maniera notevole e questo è dovuto all'utilizzo della codifica BCH.

4.4.2 Secondo scenario

Lo scenario utilizzato in questa simulazione per l'analisi del canale è il seguente:

- Estate
- Numero sorgenti = 1; Numero destinatari = 1
- Numero di raggi inviati 2000.
- Profondità acqua 130 m
- Distanza orizzontale tra sorgente e ricevitore 1500 m.

- Quota di 30m per la sorgente e 50m per il ricevitore
- Direzione invio raggi $-50 \div 6$ gradi rispetto all'orizzonte

In questo scenario il numero di raggi che sono stati rivelati sono 3 e presentano i seguenti ritardi assoluti:

- $\tau_{a0} = 0.6687966 \text{ sec}$
- $\tau_{a1} = 0.6714494 \text{ sec}$
- $\tau_{a2} = 0.6731683 \text{ sec}$

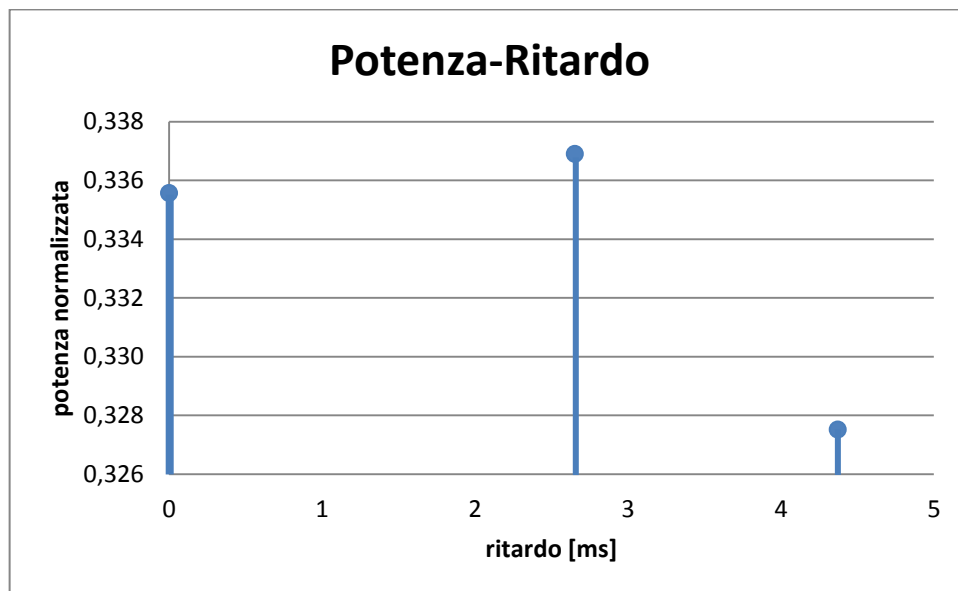


Fig. 4.5: attenuazione e ritardo introdotti su ogni cammino.

Le caratteristiche del canale sono riportate in figura 4.5, da cui si evince che i ritardi relativi sono:

- $\tau_1 = 2,6 \text{ ms}$
- $\tau_1 = 4,3 \text{ ms}$

Dal grafico si evince che la replica utilizzata dal ricevitore a singolo raggio è quella con ritardo $\tau_1 = 2,6 \text{ ms}$ mentre il ricevitore Rake le sfrutta tutte e tre. In questo caso risulta che

$6 < T_{sw} < 16 \text{ ms}$ (nella simulazione si è scelto $T_{sw} = 10$).

Lo spettro del segnale dopo l'operazione di despreading sul raggio più forte e in assenza di rumore è rappresentato in figura 4.6. Come fatto nella simulazione precedente, per determinare le frequenze istantanee bisogna calcolare i ritardi rispetto alla replica più forte e cioè quella con ritardo $\tau = 2,6 \text{ ms}$, ottenendo:

- $\tilde{\tau}_1 = 2,6 - 0 = 2,6 \text{ ms}$
- $\tilde{\tau}_2 = 4,3 - 2,6 = 1,7 \text{ ms}$

Utilizzando questi risultati, si trovano i seguenti valori di frequenza istantanea:

- $f_i^1(t) = 4160 \text{ Hz}; 11840 \text{ Hz}$
- $f_i^2(t) = 2720 \text{ Hz}; 13280 \text{ Hz}$

Anche in questo caso, come mostrato in figura 4.6, l'utilizzo di un filtro passa basso permette il recupero senza interferenza della replica.

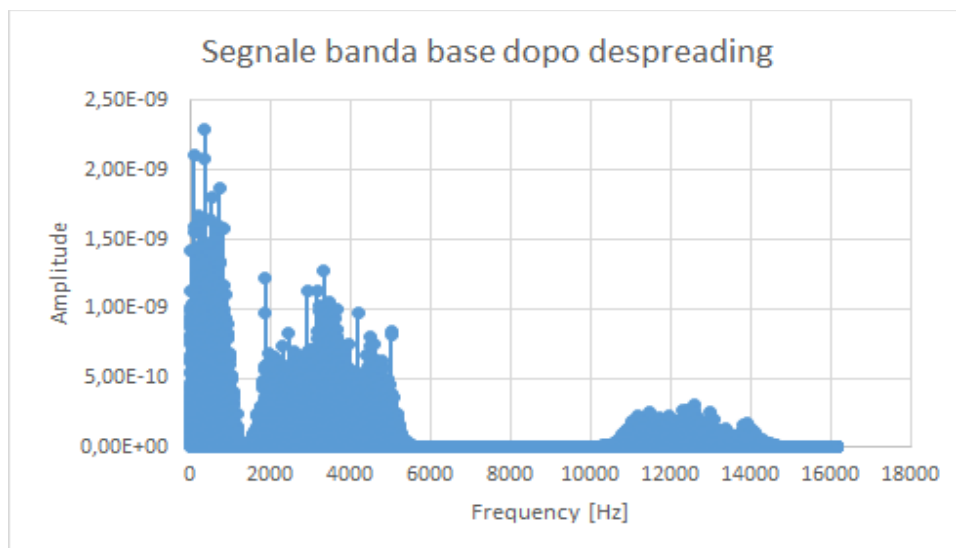


Fig. 4.6: spettro ottenuto dopo il despreading fatto sul raggio più forte

Come visto, non si è fissato un valore per N_{sr} . Infatti, in questo scenario è bene fare tre simulazioni per tre diversi valori di N_{sr} in modo da mettere in risalto l'importanza di un corretto dimensionamento del preambolo, che, come si vedrà, permette un miglioramento della BER all'aumentare di N_{sr} .

4.4.2.1 prestazioni al variare della lunghezza del preambolo

Il preambolo, come detto, è molto importante per la stima del ritardo e per la stima di canale. In particolare, all'aumentare di N_{sr} diminuisce la BER, però, questo si verifica fino ad un certo valore di N_{sr} : man mano che N_{sr} aumenta le curve di BER che si ottengono tendono ad essere simili. Infatti quando N_{sr} è sufficientemente grande, gli errori di stima del ritardo divengono banalmente piccoli da non riflettersi sulla BER in modo significativo. Inoltre, se una lunghezza troppo piccola può portare a delle stime non buone con conseguente aumento della BER, una lunghezza troppo grande fa diminuire il numero di bit informativi al secondo inviati (efficienza spettrale). Per questo, il valore di N_{sr} dipende dalle prestazioni che si vogliono ottenere in termini di BER e di efficienza spettrale.

Di seguito vengono riportati i corrispettivi grafici della BER al variare di N_{sr} , sia nel caso si utilizzi la tecnica convenzionale che quella a Rake.

- $N_{sr} = 20$

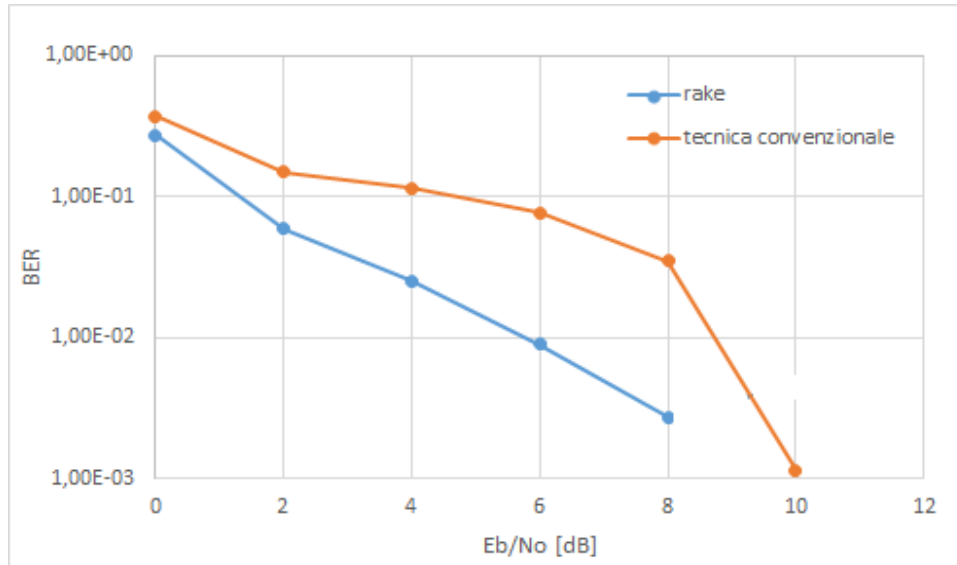


Fig. 4.7 : BER per vari valori di E_b/N_0 nel caso di ricevitore a singolo raggio e RAKE con $N_{sr} = 20$

- $N_{sr} = 40$

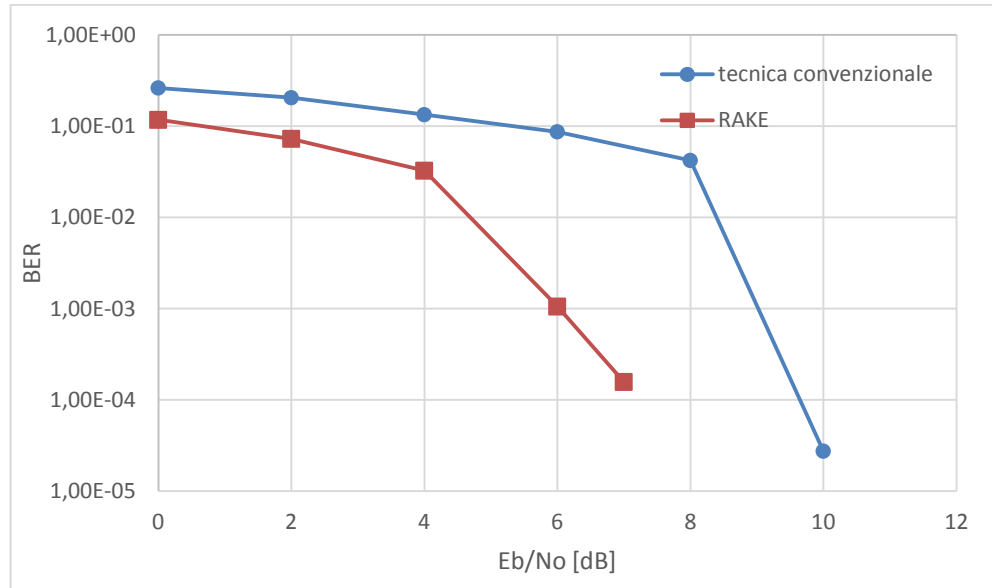
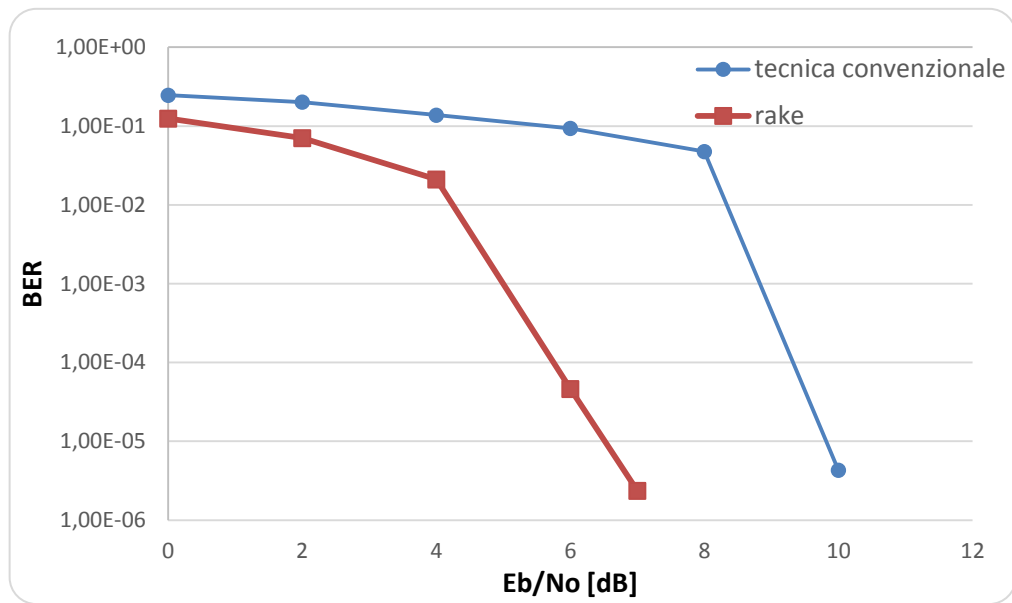


Fig. 4.8 : BER per vari valori di E_b/N_0 nel caso di ricevitore a singolo raggio e RAKE con $N_{sr} = 40$

- $N_{sr} = 80$



- Fig. 4.9: BER per vari valori di E_b/N_0 nel caso di ricevitore a singolo raggio e RAKE con $N_{sr} = 80$

Nel caso di $N_{sr} = 20$, figura 4.7, il modem non funziona correttamente in quanto gli errori di stima ritardo e stima canale sono elevati. All'aumentare di N_{sr} le prestazioni migliorano sia nel caso RAKE che con la tecnica convenzionale. Questo aumento, però, va gradualmente diminuendo all'aumentare di N_{sr} perché per valori di $N_{sr} > 80$ le curve di BER sono pressochè uguali. Aumentando N_{sr} diminuisce l'efficienza spettrale in modo ingiustificato, dato che non si ottengono miglioramenti di prestazioni.

4.4.3 Terzo scenario: condizione su T_{sw} non rispettata

Questa simulazione è stata fatta considerando sempre il secondo scenario in cui, però, variano i ritardi relativi: i valori di potenza normalizzata sono uguali mentre uno dei ritardi viene modificato. In particolare sono state realizzate due simulazioni:

1. Si fissa un valore di T_{sw} che soddisfa una delle due condizioni ma non entrambe
2. Si fissa un valore di T_{sw} che non soddisfa entrambe le condizioni

Queste simulazioni sono state realizzate per mettere in risalto il fatto che il ricevitore RAKE, anche se una delle due condizioni non è rispettata, riesce ad ottenere buone prestazioni a differenza della tecnica convenzionale che invece non funziona anche ad alti valori di E_b/N_0 .

4.4.3.1 prima simulazione

I ritardi relativi sono:

- $\tau_1 = 0,5 \text{ ms}$
- $\tau_1 = 4,3 \text{ ms}$

Da questi valori si ricava che $5 \text{ ms} < T_{sw} < 3 \text{ ms}$, quindi non esiste un valore che soddisfi entrambe le condizioni⁶. Per la simulazione si è scelto un valore di $T_{sw} = 10 \text{ ms}$ in modo che venga rispettata la condizione con il segno maggiore e $N_{sr} = 80$. Il fatto che non venga rispettata una delle due condizioni fa sì che lo spettro sulla continua venga interferito, figura 4.10, degradando le prestazioni del sistema.

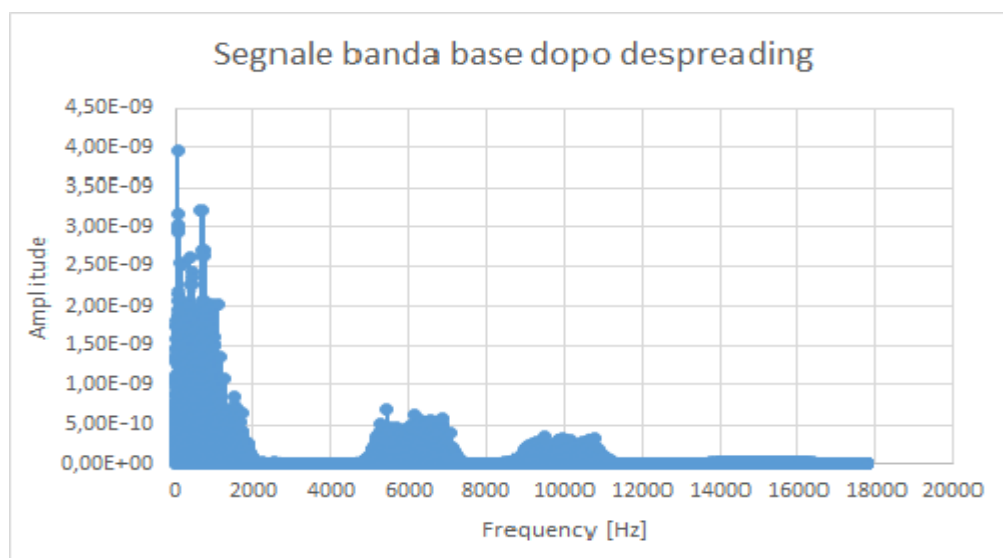


Fig. 4.10: spettro ottenuto dopo il despreading fatto sul raggio più forte

⁶ Si ricordi che M è una costante perché si è deciso di non variare i parametri di roll-off, intervallo di segnalazione e frequenze.

La replica più forte è quella a $\tau = 0,5 \text{ ms}$ e quindi i ritardi da considerare sono

- $\tilde{\tau}_1 = 0,5 - 0 = 0,5 \text{ ms}$
- $\tilde{\tau}_2 = 4,3 - 0,5 = 3,8 \text{ ms}$.

I valori di frequenza istantanea che si trovano sono:

- $f_i^1(t) = 800 \text{ Hz}; 15200 \text{ Hz}$
- $f_i^2(t) = 6080 \text{ Hz}; 9920 \text{ Hz}$

Questi valori ottenuti in teoria, possono essere confrontati con il grafico ottenuto per simulazione, figura 4.10: la banda dello spettro sulla continua, in teoria, dovrebbe essere di 1300 Hz, mentre si ha una banda di circa 2000 Hz. Questo è dovuto al fatto che lo spettro associato alla replica con ritardo $\tilde{\tau}_1$ ha una frequenza centrale di 800 Hz ($800+1300 = 2100 \text{ Hz}$) che quindi va a sovrapporsi provocando interferenza. In questa situazione si hanno delle ripercussioni notevoli soprattutto quando si utilizza la tecnica convenzionale. Infatti, come mostrato in figura 4.11, nel caso della tecnica convenzionale il sistema non funziona neanche per alti valori di E_b/N_0 . Invece, nel caso del ricevitore RAKE, anche se le prestazioni degradano rispetto al caso visto nel *secondo scenario*, si riesce a far funzionare il sistema: con il ricevitore RAKE sfruttiamo tutte le repliche, quindi anche quella con ritardo $4,3 \text{ ms}$.

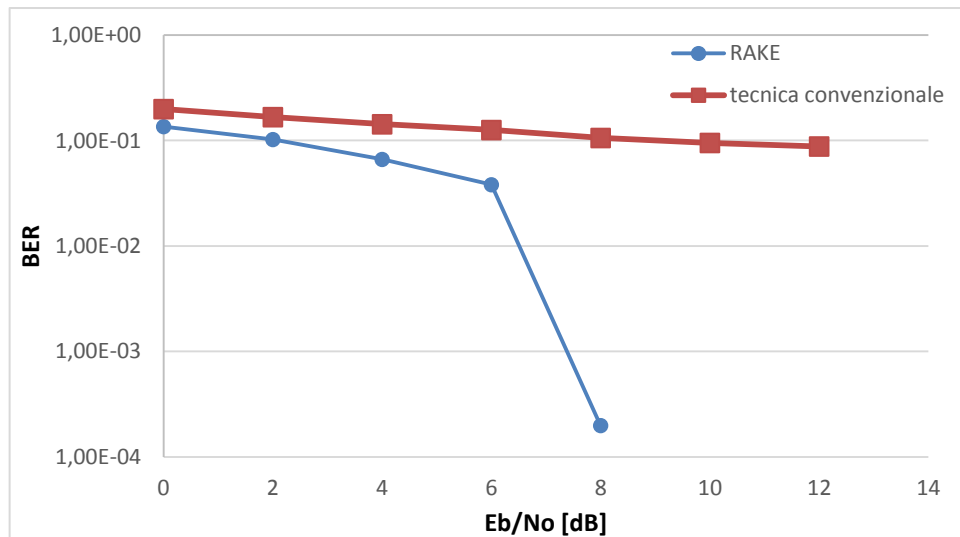


Fig. 4.11: BER nel caso in cui T_{sw} non soddisfi una delle due condizioni.

Infatti, se si guarda lo spettro in uscita al despreader che si sincronizza sulla replica con ritardo 4.3 ms , figura 4.12, lo spettro sulla continua ha una banda di 1300 Hz e quindi non si ha interferenza. In tal caso si può utilizzare il filtro passa basso per recuperare la replica.

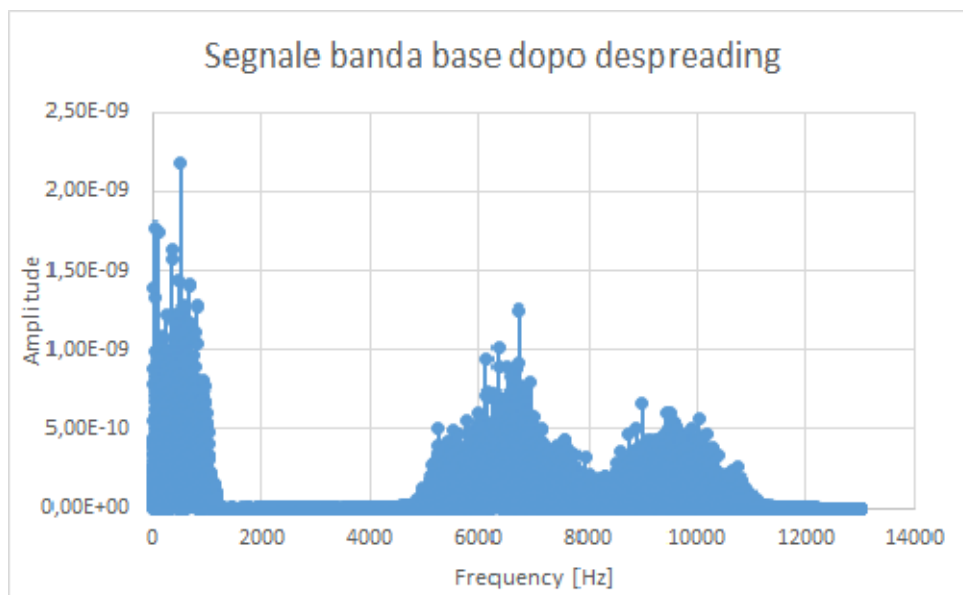


Fig. 4.12: spettro ottenuto dopo il despreading fatto sulla replica con ritardo $\tau_1 = 4,3\text{ ms}$.

4.4.3.2 seconda simulazione

Ora, supponiamo di prendere un valore di T_{sw} che non soddisfi entrambe le condizioni, ad esempio $T_{sw} = 4$. In questa situazione neanche il ricevitore a RAKE funziona in quanto tutti gli

spettri interferiscono fra di loro. Infatti, come rappresentato in figura 4.13, anche per alti valori di E_b/N_0 la BER assume valori elevati.

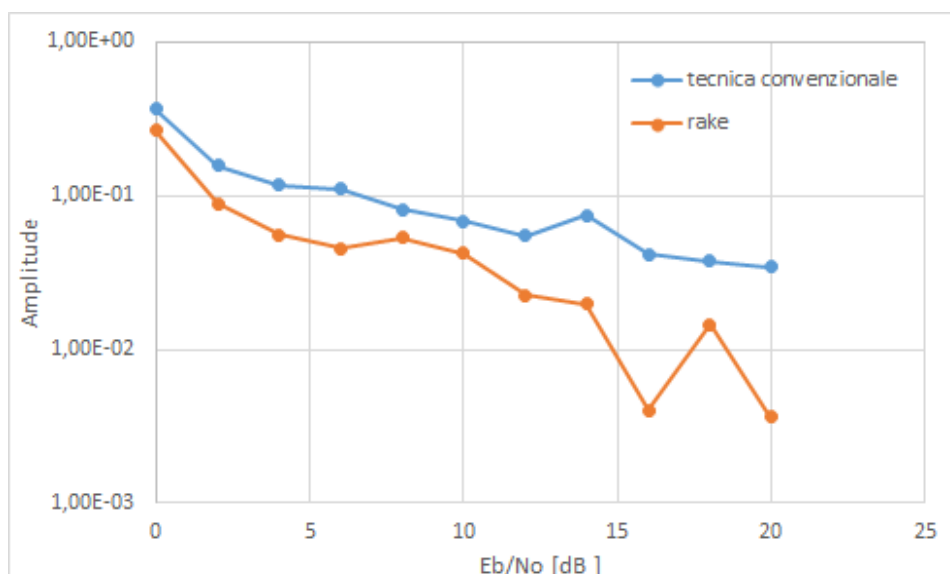


Fig. 4.13: BER nel caso in cui T_{sw} non soddisfi alcuna delle due condizioni

Questo ci fa capire quanto sia importante settare in maniera opportuna il parametro T_{sw} .

4.4.4 Quarto scenario

In questa simulazione, lo scenario utilizzato è analogo a quello visto nel *secondo scenario* con la differenza che il periodo è inverno.

- Inverno
- Numero sorgenti = 1; Numero destinatari = 1
- Profondità acqua 130 m
- Distanza orizzontale tra sorgente e ricevitore 1500 m.
- Quota 30 m per la sorgente e 50 m per il ricevitore
- Direzione invio raggi $-50 \div 6$ gradi rispetto all'orizzonte

In questo scenario il numero di raggi che sono stati rivelati sono 6 e presentano i seguenti ritardi assoluti:

- $\tau_{a0} = 0.66936123 \text{ sec}$
- $\tau_{a1} = 0.67227638 \text{ sec}$
- $\tau_{a2} = 0.70632064 \text{ sec}$
- $\tau_{a3} = 0.71557277 \text{ sec}$
- $\tau_{a4} = 0.78091115 \text{ sec}$
- $\tau_{a5} = 0.79512405 \text{ sec}$

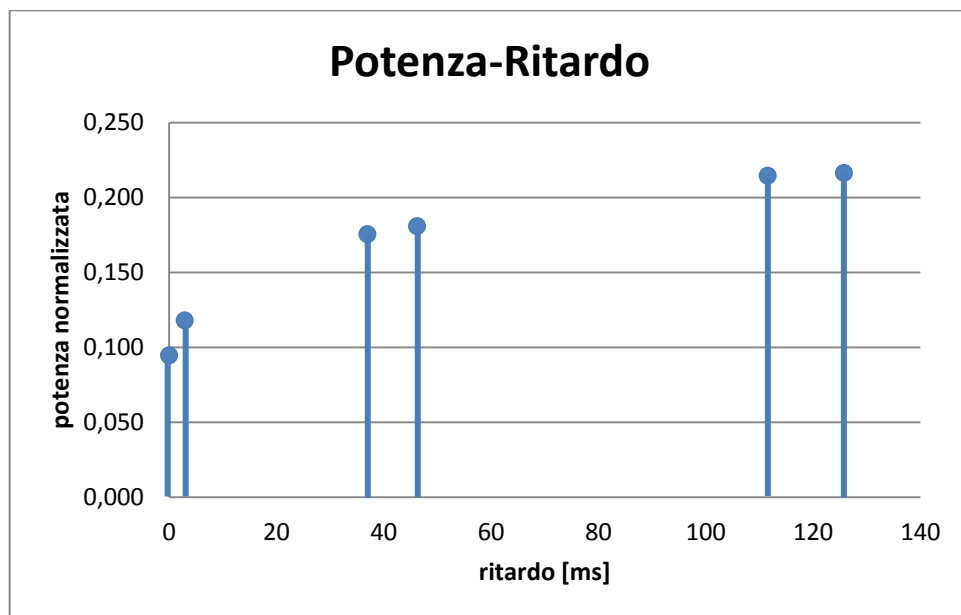


Fig. 4.14: attenuazione e ritardo introdotti su ogni cammino.

Le caratteristiche del canale sono riportate in figura 4.14, da cui si evince che i ritardi relativi sono:

- $\tau_1 = 2,9 \text{ ms}$
- $\tau_2 = 37 \text{ ms}$
- $\tau_3 = 46,2 \text{ ms}$
- $\tau_4 = 111,5 \text{ ms}$
- $\tau_5 = 126 \text{ ms}$

In questo scenario, il ricevitore a singolo raggio sfrutta il raggio con ritardo 126 ms o quello con ritardo 112 ms, mentre il Rake utilizza le repliche con ritardi di 46 ms, 112 ms e 126 ms (si ricordi che il Rake ha 3 rami e, quindi, può sfruttare fino ad un massimo di 3 repliche). Questa situazione è peggiore di quella del *secondo scenario*. Infatti, se calcoliamo le condizioni su T_{sw} troviamo che $151 < T_{sw} < 18 \text{ ms}$. Quindi non esiste un valore che soddisfi entrambe le condizioni. Inoltre ci sono 3 repliche in più che vanno ad interferire.

Prendendo un valore di $T_{sw} = 180 \text{ ms}$ e $N_{sr} = 360$ se si visualizza il grafico relativo alla BER, figura 4.15, si nota che la tecnica convenzionale non funziona neanche per alti valori di E_b/N_0 a differenza del RAKE in cui la BER inizia ad assumere valori accettabili a partire da $E_b/N_0 = 10 \text{ dB}$.

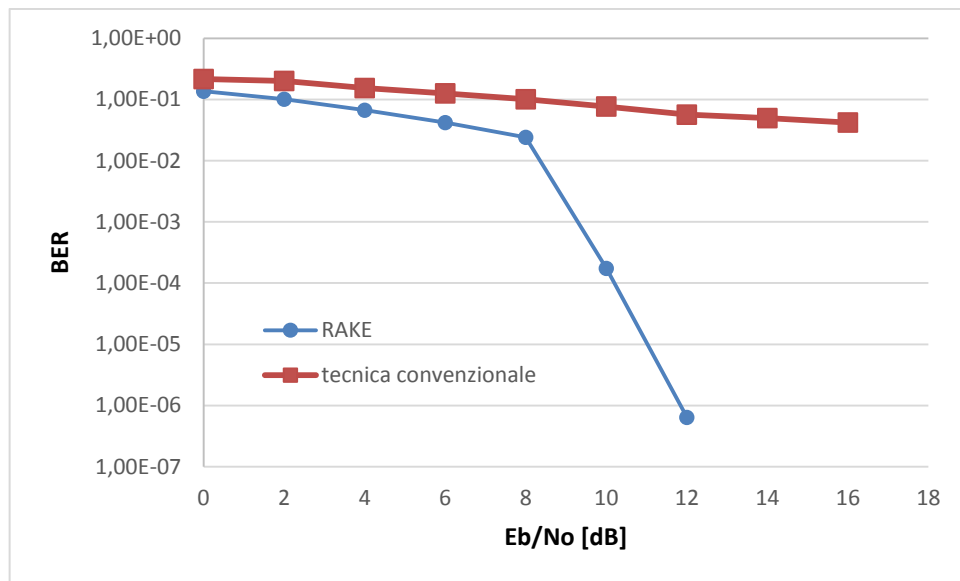


Fig. 4.15: BER per vari valori di E_b/N_0 nel caso di ricevitore a singolo raggio e RAKE

Andando a fare un confronto tra questo scenario e il terzo scenario in cui T_{sw} non è rispettato, si evince che le prestazioni sono migliori nel terzo scenario e questo è da imputare al fatto che ci sono più cammini e che questi introducono ritardi sparsi su di un grande range $[2.9 \div 126]$ che comporta dei valori di frequenza istantanea che si vanno a sovrapporre alla continua.

Dalle simulazioni fatte si possono dedurre 2 concetti importanti:

1. Bisogna settare in maniera opportuna il valore di N_{sr} ma soprattutto quello di T_{sw} .
2. L'utilizzo del RAKE è da preferire, perché ha delle prestazioni migliori in termini di BER ed, inoltre, permette il corretto funzionamento del sistema anche se una delle condizioni su T_{sw} non è verificata.

CAPITOLO 5: PROTOTIPO TRASMETTITORE MODEM ACUSTICO

5.1 Introduzione

In quest'ultimo capitolo viene descritto come integrare nel sistema le funzioni DAQmx per fare in modo che il segnale a spettro espanso, generato via software, venga trasmesso dalla scheda di acquisizione/generazione trattata nel capitolo 1. Inoltre, per verificare che il segnale è stato effettivamente trasmesso, si è utilizzato un analizzatore di spettro.

5.2 Codice per la gestione della scheda

Come riportato nel capitolo 2, in LabView esiste un set predefinito di funzioni che servono per la gestione di acquisizione e generazione dati attraverso una scheda. Queste funzioni sono le stesse per tutte le famiglie di dispositivi di acquisizione/generazione e, quindi, si è ritenuto utile mostrare quale possa essere una possibile implementazione base di tali funzioni per la gestione della scheda, figura 5.1.

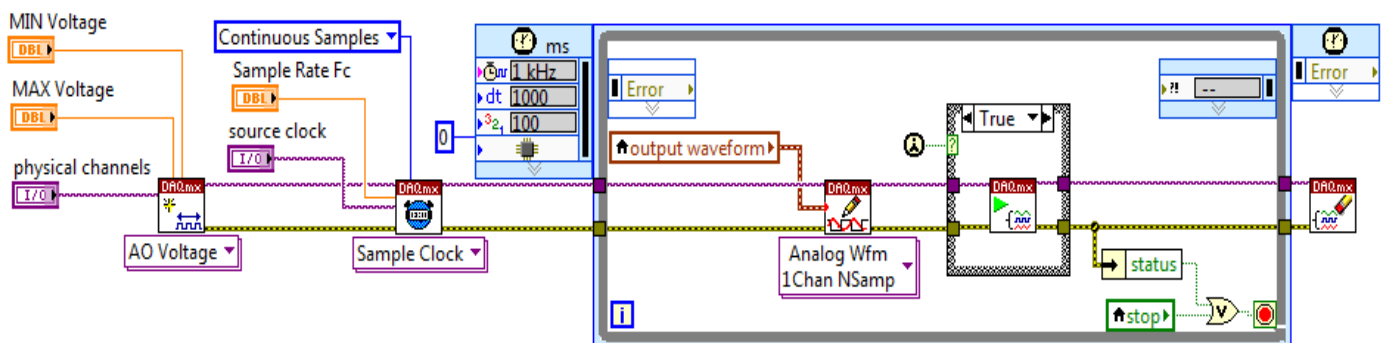


Fig. 5.1: possibile implementazione delle funzioni DAQmx per la gestione della scheda

Queste funzioni sono state già trattate singolarmente nel capitolo 2, e quindi, di seguito ne viene data solo una breve descrizione. Partendo da sinistra, figura 5.1, viene creato il canale fisico sul quale scrivere i dati; una volta creato, bisogna gestire la temporizzazione ed in particolare settare la frequenza di campionamento e la sorgente del clock (in questo caso si è utilizzata quella dello chassis); a questo punto viene passato, attraverso una variabile locale, il segnale a spettro espanso

nel dominio del tempo alla funzione *write*, che si occuperà di scrivere la forma d'onda in un task; per poter eseguire tale compito, bisogna che il task sia nello stato di *running* e questo viene fatto dalla funzione *start* eseguita solo al primo ciclo; infine, quando la trasmissione del segnale è conclusa si cancella il Task e si rilasciano tutte le risorse occupate.

Una considerazione va fatta sulla struttura *Timed Loop* che racchiude la funzione *write* e *start*. Questa è concettualmente analoga al *while* (permette di eseguire più volte una parte di codice) con la possibilità, però, di implementare alcune funzionalità: gestione dei core, gestione dei tempi di esecuzione del ciclo ecc. Ad esempio si consideri la gestione dei core: come già detto, LabView funziona in maniera “parallela” e quindi è possibile, ad esempio, sfruttare due cicli *while* in modo da eseguire contemporaneamente la generazione del segnale e la sua scrittura su scheda. Ovviamente la parallelizzazione è virtuale, però, su un sistema multicore è possibile, tramite il *timed loop*, indicare quale parte di codice eseguire su un determinato core: l'etichetta '0', figura 5.1, collegata al simbolo del processore nel *Timed Loop*, sta a indicare che la parte di codice racchiusa nel *Timed Loop* viene eseguita sul core '0'. Quindi se si crea un altro *Timed Loop* in cui inserire il codice per la generazione del segnale e si etichetta con il numero '1', allora il sistema effettivamente funziona in maniera parallela in quanto una parte di codice è eseguita sul core '0' e l'altra sul core '1'.

5.3 Hardware utilizzato nel progetto

In figura 5.2 è visualizzato l'hardware usato nel progetto. Partendo da sinistra, si ha lo chassis comprensivo di controller e scheda di acquisizione, una basetta (in blu) e l'analizzatore di spettro. Il controller elabora i dati generati via software e li passa alla scheda di acquisizione che li trasmette su un'uscita analogica selezionata tramite la funzione *AO voltage*; questo segnale, attraverso un cavo, viene inviato ad una basetta dalla quale è possibile prelevare e inviarlo ad un analizzatore di spettro per verificare che sia quello corrispondente ad un segnale a spettro espanso come ottenuto dopo una modulazione S2C.

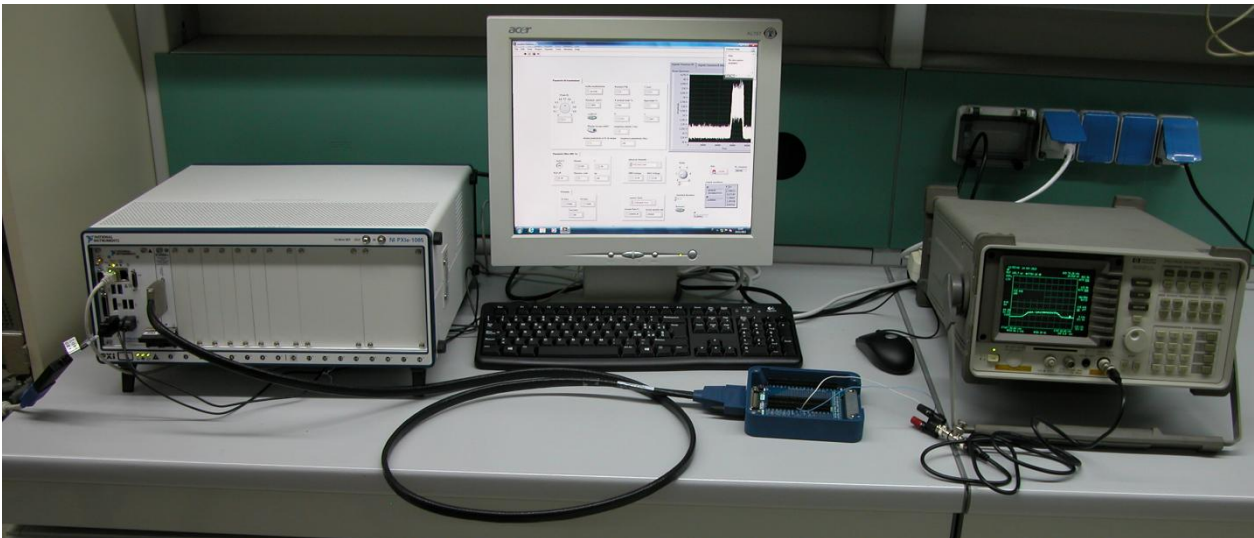


Fig. 5.2: componenti hardware utilizzati nel progetto

In figura 5.3 è mostrato lo chassis NI-PXIe-1085 con alloggiati al suo interno il controller NI PXIe-8135 e la scheda di acquisizione/generazione dati NI PXIe-6361 di cui si è già parlato nel capitolo 1. Come detto, la scheda presenta svariati ingressi e uscite analogiche e linee digitali, ognuno dei quali è associato ad un determinato pin. Come si vede dall'immagine, alla scheda 6361 è collegato un cavo che serve per far transitare il segnale elettrico generato ed inviarlo in ingresso alla basetta, figura 5.4.



Fig. 5.3: chassis con controller e scheda di acquisizione

Questa basetta presenta degli alloggiamenti, contrassegnati da numeri, dai quali è possibile prelevare il segnale generato e/o acquisire un segnale per poi inviarlo alla scheda. Ad esempio, i due fili (bianco e blu) nell'immagine sono utilizzati per prelevare il segnale inviato sull'uscita AO 0: il filo blu preleva il segnale elettrico dal terminale 22, mentre quello bianco è la massa associata ad AO 0 corrispondente al terminale 55.

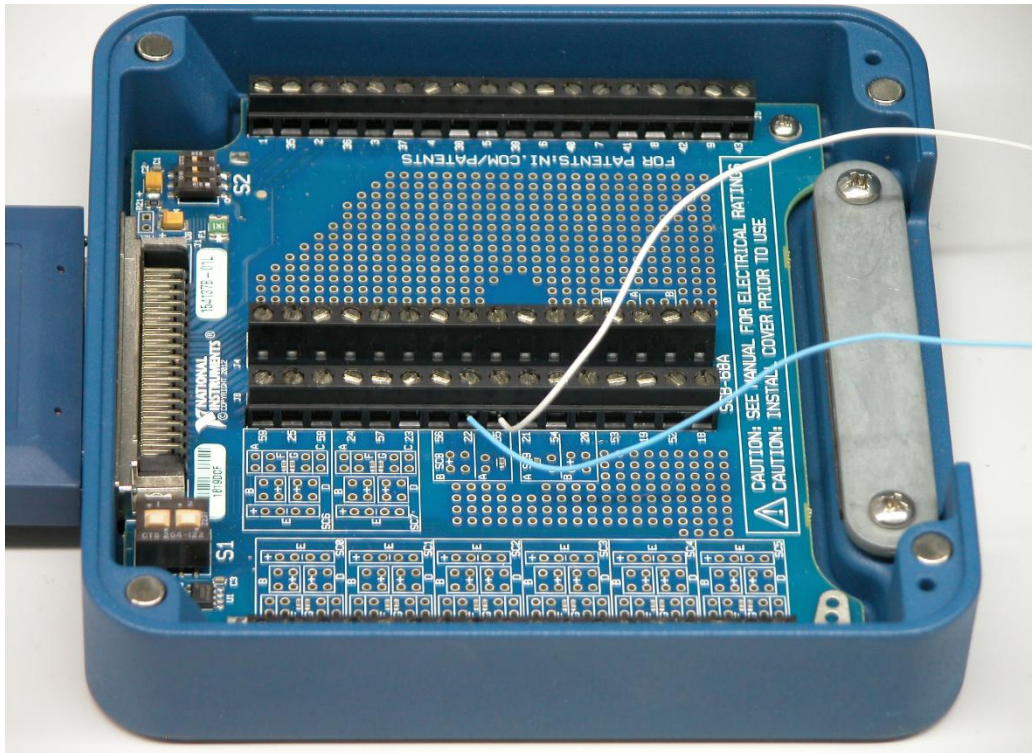


Fig. 5.4: basetta per prelevare o acquisire segnali

Infine, questi fili vengono collegati a due connettori in modo da inviare il segnale ad un analizzatore di spettro, che visualizza il segnale a RF a spettro espanso della modulazione S2C (figura 5.5).

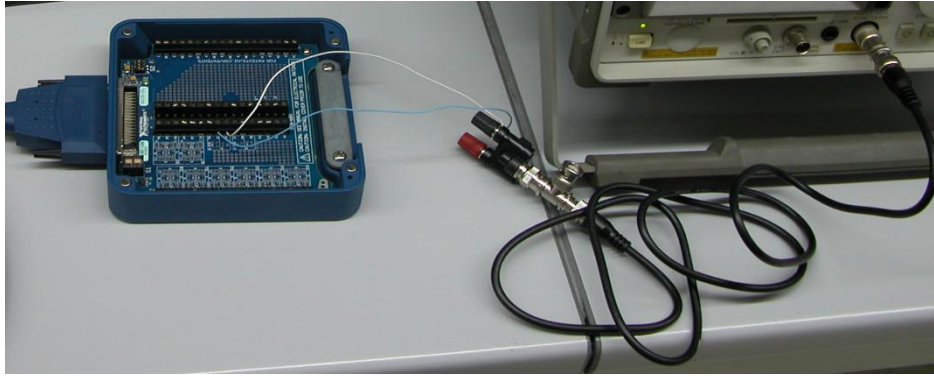


Fig. 5.5: passaggio del segnale all'analizzatore di spettro

Per completezza viene mostrato anche il Pinouts/Frontal Connector che indica a cosa corrispondono i numeri presenti sulla basetta.

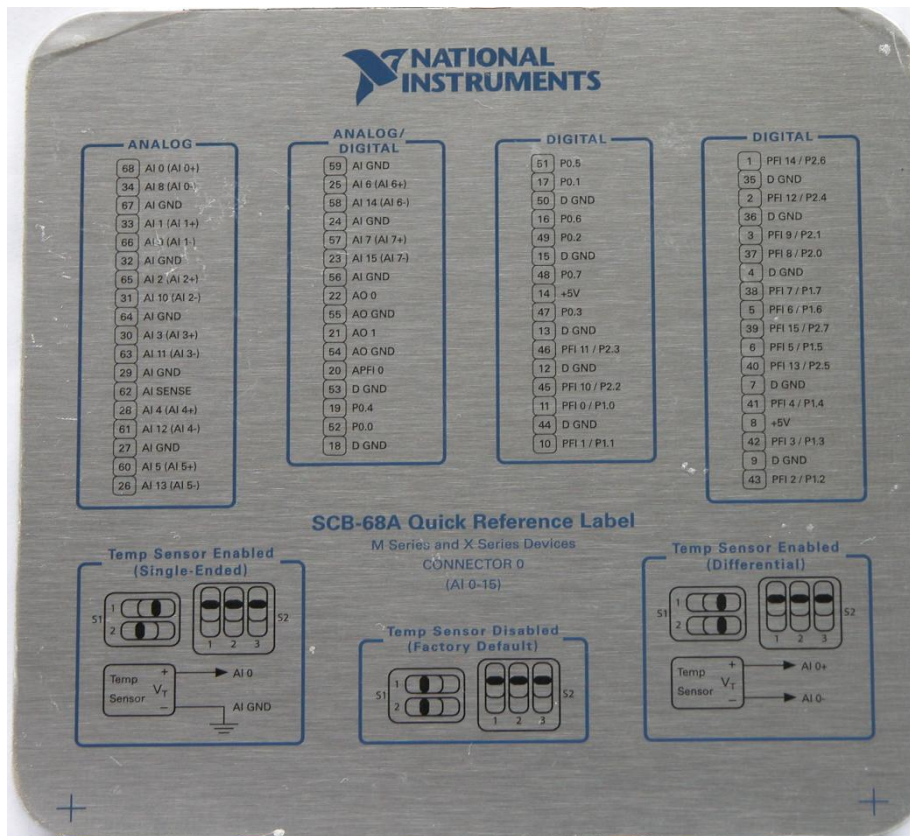


Fig. 5.6: Pinouts/Frontal Connector della basetta

5.3.1 Misure dello spettro

Una volta che l'hardware è stato collegato è utile effettuare delle misure per testare se il sistema funziona correttamente. Le misure sono state effettuate utilizzando un analizzatore di spettro (figura 5.7).

Per capire al meglio le immagini riportate di seguito è utile fare due considerazioni:

1. Nei capitoli precedenti si è detto che le frequenze usate nel modem sono tra 18 kHz e 34 kHz. Invece lo spettro visualizzato nell'immagine va da 57 kHz a 73 kHz. Questo perché lo strumento utilizzato ha un limite inferiore di banda di 9kHz (riesce a fare l'analisi spettrale di segnali da 9 kHz a 1.8 GHz) e quindi si è preferito non lavorare vicino al limite inferiore di banda.
2. Lo spettro visualizzato è il risultato di una media mobile fatta su più spettri (il numero di medie fatte è indicato sulla sinistra dell'immagine con la sigla AVG XX).

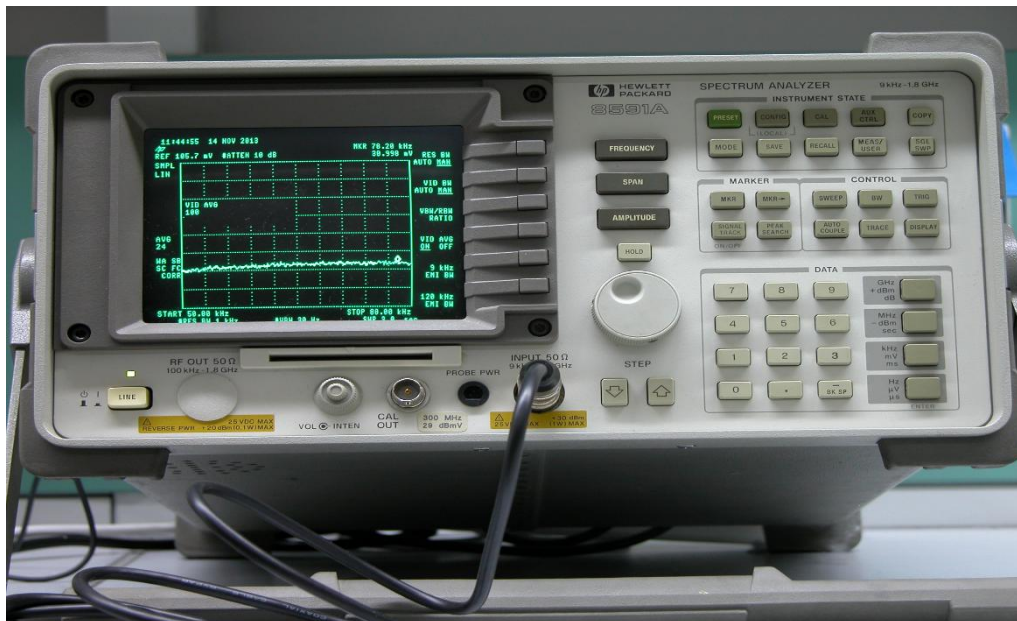


Fig. 5.7: analizzatore di spettro utilizzato per le misure.

Una prima misura è stata fatta considerando solo rumore AWGN (figura 5.8). Come si vede nell'immagine lo spettro è piatto nel range di frequenze considerato.

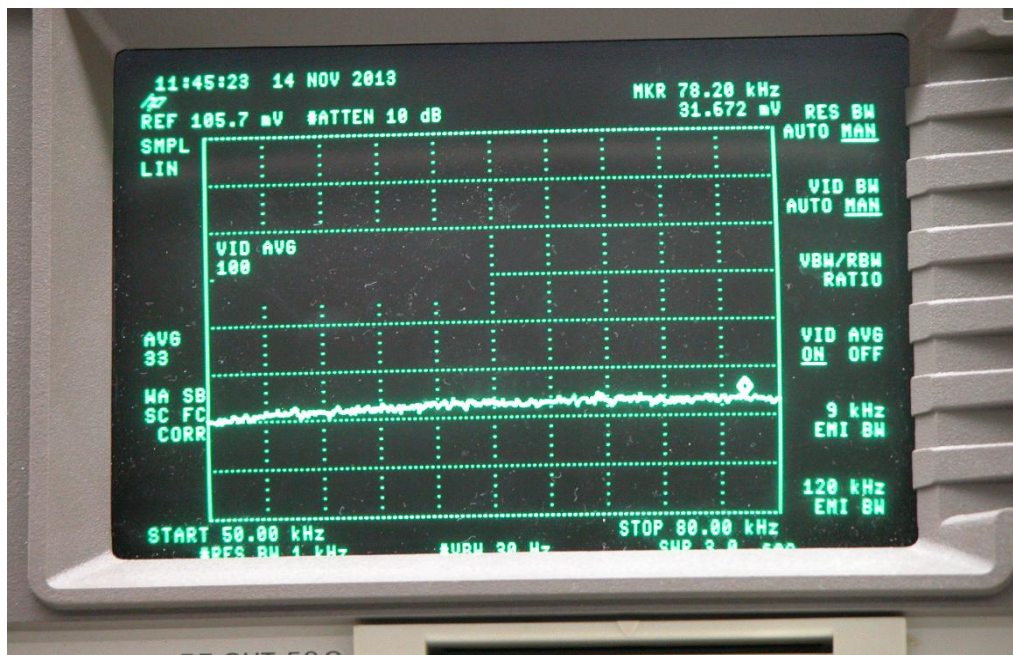


Fig. 5.8: spettro rumore AWGN

Nella seconda misura si è inviato all'analizzatore il solo segnale, figura 5.9, quindi in assenza di rumore. Dall'immagine si vede che lo spettro rispetta i valori di frequenza utilizzati: $57 \div 73$ kHz



Fig. 5.9: spettro segnale in assenza di rumore

Nella terza misura si è calcolato lo spettro di un segnale in presenza di rumore (figura 5.10). Nell'immagine si vede che lo spettro è quasi al livello del rumore, quindi diventa difficile per un utente capire se vi è un segnale utile o se è solo rumore.

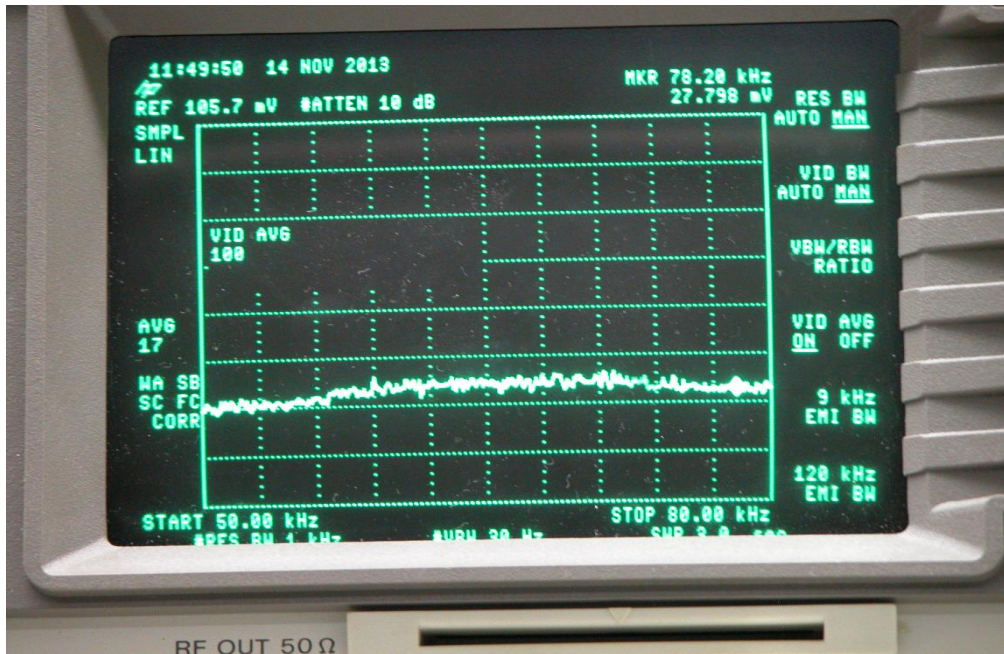


Fig. 5.10: spettro segnale in presenza di rumore

Nella quarta e ultima misura si è aumentata la potenza di segnale lasciando invariata la potenza di rumore (figura 5.11). Dall'immagine si vede che, rispetto alla terza misura, in questo caso il segnale utile è riconoscibile.

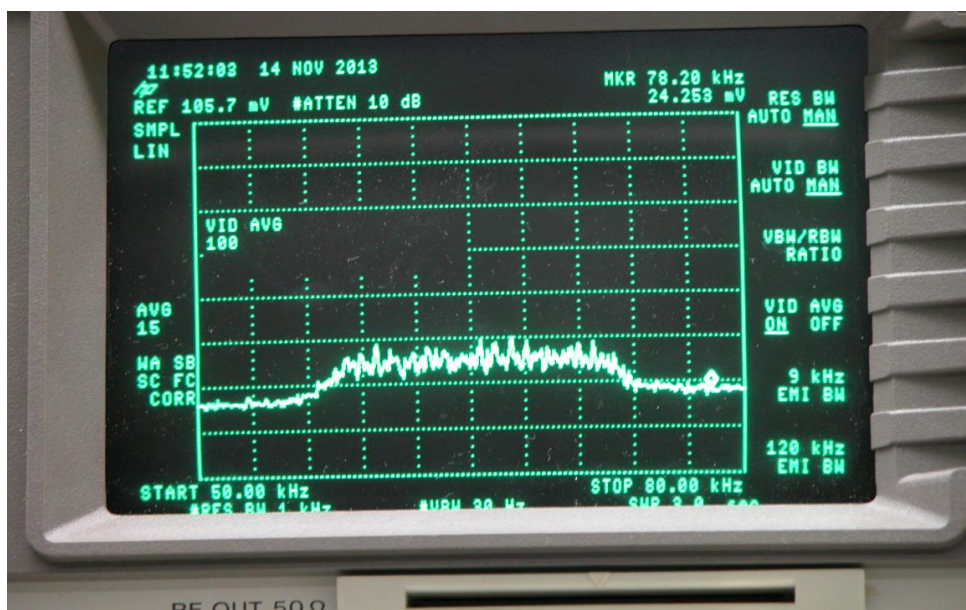


Fig. 5.11: spettro segnale in presenza di rumore

SITOGRAFIA

- 1) National Instruments, **Learn 10 Functions in NI-DAQmx**
<http://www.ni.com/white-paper/2835/en#toc10>
- 2) National Instruments, **NI X Series Multifunction Data Acquisition**
<http://sine.ni.com/ds/app/doc/p/id/ds-151/lang/it>
- 3) National Instruments, **NI PXIe-8135**
<http://sine.ni.com/nips/cds/view/p/lang/it/nid/210545>
- 4) National Instruments, **NI PXIe-1085**
<http://sine.ni.com/nips/cds/view/p/lang/it/nid/210863>
- 5) National Instruments, **NI-DAQmx Software**
<http://sine.ni.com/nips/cds/view/p/lang/it/nid/10181>
- 6) National Instruments, **Channels and Tasks in NI-DAQmx**
<http://zone.ni.com/reference/en-XX/help/370466V-01/TOC7.htm>